

# A Refined Second-order Arnoldi (RSOAR) Method for the Quadratic Eigenvalue Problem and Implicitly Restarted Algorithms\*

Zhongxiao Jia<sup>†</sup>

Yuquan Sun<sup>‡</sup>

## Abstract

To implicitly restart the second-order Arnoldi (SOAR) method proposed by Bai and Su for the quadratic eigenvalue problem (QEP), it appears that the SOAR procedure must be replaced by a modified SOAR (MSOAR) one. However, implicit restarts fails to work provided that deflation takes place in the MSOAR procedure. In this paper, we first propose a Refined MSOAR (abbreviated as RSOAR) method that is based on the refined projection principle. We derive upper bounds for residual norms of the approximate eigenpairs obtained by the MSOAR and RSOAR methods. Based on them, we propose a reliable tolerance criterion for numerical breakdown that makes the MSOAR and RSOAR methods converge to a prescribed accuracy. This criterion also serves to decide numerical deflation. We consider the central issue of selecting the shifts involved when implicitly restarting the MSOAR and RSOAR algorithms. We propose the exact and refined shifts for the two algorithms, respectively, and present an effective approach to treat the deflation issue in implicit restarts, so that the implicit restarting scheme works unconditionally. Numerical examples illustrate the efficiency of the restarted algorithms and the superiority of the restarted RSOAR to the restarted MSOAR.

**Keywords.** Ritz value, Ritz vector, refined Ritz vector, implicit restart, exact shifts, refined shifts, numerical deflation, numerical breakdown.

**AMS Subject Classification (2000).** 65F15, 15A18

## 1 Introduction

Consider the large quadratic eigenvalue problem (QEP)

$$(\lambda^2 M + \lambda C + K)x = 0, \quad (1)$$

where  $M$ ,  $C$ ,  $K$  are  $n \times n$  real or complex matrices with  $M$  nonsingular. QEP's arise in many applications, see, e.g., [2, 27]. We are interested in a few largest eigenvalues in magnitude or a few eigenvalues nearest to a target in the complex plane. A commonly used approach is to linearize the QEP. There are a number of linearizations available [27]. For example, if  $M$

---

\*Supported by National Basic Research Program of China 2011CB302400 and the National Science Foundation of China (No. 11071140).

<sup>†</sup>Department of Mathematical Sciences, Tsinghua University, Beijing 100084, People's Republic of China, jjazx@tsinghua.edu.cn

<sup>‡</sup>LMIB & School of Mathematics and Systems Science, BeiHang University, Beijing 100191, People's Republic of China, sunyq@buaa.edu.cn

is invertible, one of the most often used linearizations is to transform it to the generalized eigenvalue problem

$$\begin{bmatrix} -C & -K \\ I & 0 \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix} = \lambda \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix}, \quad (2)$$

which can be further reduced to the standard linear eigenvalue problem

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix} = \lambda \begin{bmatrix} \lambda x \\ x \end{bmatrix}, \quad (3)$$

where  $A = -M^{-1}C$ ,  $B = -M^{-1}K$ . Note that (3) corresponds to the monic QEP

$$(\lambda^2 - \lambda A - B)x = 0. \quad (4)$$

The mathematical theory on (2) and (3) has been well established and a number of numerical methods have been available for solving it [1, 6, 23, 25, 28]. A serious drawback of the linearization is that general numerical methods do not take the structures of (2) and (3) into account, so that resulting projected matrices do not possess the same structures as those of the original problems. Moreover, essential structures of the QEP, such as possible symmetries of  $M$ ,  $C$  and/or  $K$  are not preserved. For example, if  $M$ ,  $C$  and  $K$  are real and  $M$  is symmetric positive definite,  $K$  is symmetric and  $C$  is skew-symmetric, then the eigenvalues of the QEP appear in pairs  $(\lambda, -\lambda)$  if  $\lambda$  is real or purely imaginary and in quadruplets  $(\lambda, -\lambda, \bar{\lambda}, -\bar{\lambda})$  if  $\lambda$  is complex. However, the eigenvalues of projected matrices for the linearized problems do not have such spectrum structures. Therefore, it is essential to work on the QEP directly. We will develop methods that can not only preserve essential structures and properties of the QEP but also take some advantages of the Arnoldi type methods for the linearized problem.

The Second-Order Arnoldi (SOAR) method proposed by Bai and Su [3] is a Rayleigh–Ritz method that works on (1) directly. They propose a SOAR procedure that first computes an orthonormal basis for a second-order Krylov subspace generated by the matrices  $A$  and  $B$  simultaneously in an elegant way and then projects (1) onto this subspace to get a projected QEP that preserves the physical structure of the original QEP. They have established some relationships between the SOAR method and the Arnoldi method for the linear problem (3) and shown that it has some merits of the Arnoldi method. The SOAR procedure is also used in dimension reduction of large scale second-order systems [4]. It preserves the second-order structure and achieves the same order of approximation as the standard Arnoldi method via linearization. A unified general convergence theory has recently been established in [9], which has generalized some of the results on the Rayleigh–Ritz method for the linear eigenvalue problem. It is proved that for a sequence of projection subspaces containing increasingly accurate approximations to the desired eigenvector there is a Ritz value that converges to the desired eigenvalue while the Ritz vectors converge conditionally and may fail to converge. Alternatively, we can compute a refined Ritz vector whose unconditional convergence is guaranteed.

Due to the storage requirement and/or computational cost, to be practical, it is generally necessary to restart the SOAR method. Unfortunately, it appears that the implicit restarting technique [24] is not applicable to the SOAR method. This is a major disadvantage. Meerbergen [21] proposes a Quadratic Arnoldi (Q-Arnoldi) method that is an alternative of the SOAR method. The Q-Arnoldi method exploits the structure of the linear problem to reduce the memory requirements by about a half and can compute a partial Schur form of the underlying linearization with respect to the structure of the Schur vectors, so that the implicit restarting technique can be applied. Otto [22] considers some theoretical and

numerical aspects of the SOAR method. Since implicit restarting cannot be applied to the SOAR method, he proposes a modified second-order Arnoldi (MSOAR) procedure that replaces the original special starting vector by a general one. Based on the MSOAR procedure and under the restrictive assumption that there is *no* deflation in the MSOAR procedure, he has developed an implicitly restarted MSOAR algorithm. It is known from [3, 22] that deflation or breakdown may take place in the SOAR and MSOAR procedures but deflation has a completely different consequence from that of breakdown. Bai and Su [3] proves that the SOAR method will find some exact eigenpairs of QEP (1) if breakdown occurs but no eigenpair is found if deflation takes place. A remedy strategy is given in [3] that can continue the SOAR procedure until the SOAR method converges. The strategy has been adapted to the MSOAR procedure [22]. However, it is not clear what will happen when deflation occurs in the MSOAR procedure. Relationships are also not known between the modified second-order Krylov subspace and its related standard Krylov subspace and between the MSOAR procedure and the standard Arnoldi process. As in the SOAR method, it turns out that similar relationships are basic for understanding the MSOAR methods.

There are explicit and elegant expressions on the residual norms of approximate eigenpairs and the Arnoldi process, which are used to design reasonable and reliable criteria for numerical breakdown [23, 25, 28]. However, no results have been established for residual norms of the approximate eigenpairs. Such kind of explicit expressions are certainly appealing and extremely important to better understand and implement the SOAR and MSOAR methods. Without them, one would not be able to propose reliable and reasonable criteria for deciding numerical breakdown. Such kind of results also appear to play a vital role in designing a reasonable tolerance criterion for numerical breakdown and deflation.

There are a number of unsolved important problems in the implicitly restarted MSOAR algorithm. As mentioned above, the implicitly restarted algorithms proposed by Otto [22] critically requires that no deflation occurs in the MSOAR procedure. Whenever deflation takes place, implicit restarting fails completely and cannot work. This requirement limits the applicability and generality of the algorithms. Another central problem is reasonable selection of the shifts involved. Otto [22] proposes exact second-order shifts for use within the algorithm. Unlike those implicitly restarted algorithms for the linear eigenvalue problem, it is distinctive that candidates for shifts exceed the number of shifts allowed. This makes a selection of shifts more subtle and complicated than that for implicitly restarted Arnoldi type algorithms for linear eigenvalue problems. Otto does not show how to reasonably select 'correct' shifts among the candidates. As it will appear, a reasonable selection of shifts is mathematically nontrivial, and we must be careful and do more work.

In this paper, combining the SOAR and MSOAR procedures with the refined projection principle [9, 11] (see also [1, 25, 28]), we propose a refined second-order Arnoldi (RSOAR) method that can guarantee the unconditional convergence of refined Ritz vector when a subspace is accurate enough [9]. We will further investigate second-order Krylov subspaces and highlight on the deflation and breakdown issues in the MSOAR procedure. We establish some basic connections between the second-order Krylov subspaces and related standard Krylov subspaces, showing the equivalence between them as well as between the MSOAR method and the standard Arnoldi method. We prove that some exact eigenpairs are found when breakdown occurs in the MSOAR procedure. We also prove that when deflation takes place the first time at some step in the MSOAR procedure, the standard Arnoldi process must do not break down. We the focus on criteria on numerical deflation and breakdown. Such criteria are extremely important and serve for two purposes: one of them is to ensure the convergence of the MSOAR and RSOAR methods when numerical breakdown occurs; the other is for the numerical stability of the MSOAR procedure. To this end, we derive upper bounds for residual norms of the approximate eigenpairs obtained by the MSOAR and

RSOAR methods and establish some important relationships between them and numerical breakdown. Based on these results, we propose a reasonable and reliable tolerance criterion for deciding numerical deflation and breakdown.

After the above is carried out, we consider two key issues when implicitly restarting the MSOAR and RSOAR algorithms: selection of shifts and treatment of numerical deflations. Just as the mechanism for those implicitly restarted Krylov subspace algorithms for linear eigenvalues problems and SVD problems [12, 14, 17, 18, 24], it turns out that a proper selection of shifts involved is one of the keys for success of the implicitly restarted algorithms. Based on the results on the equivalence between the second-order Krylov subspaces and the standard Krylov subspace as well as the MSOAR procedure and the standard Arnoldi process, we will see how shifts should be chosen. We propose *new* exact shifts that are different from Otto's and the refined shifts for respective use within the implicitly restarted MSOAR and RSOAR algorithms. The refined shifts are based on the refined Ritz vectors and theoretically better than the exact shifts. As was pointed out previously, now both exact and refined shift candidates are more than the shifts allowed. This makes a selection of shifts subtle and complicated. We show how to reasonably select the desired shifts among them for each algorithm. We present an efficient algorithm to compute the exact and refined shift candidates reliably. Finally, we consider the critical issue of how to realize implicit restarts when encountering numerical deflations. We propose an effective approach to cure numerical deflations, so that implicitly restarted MSOAR and RSOAR algorithms are of general applicability and can be run unconditionally.

The rest of this paper is organized as follows. In Section 2 we review second-order Krylov subspaces and the SOAR and MSOAR procedures and describe the SOAR and MSOAR methods. We establish a number of basic and important properties that connect second-order Krylov subspaces to standard Krylov subspaces and the MSOAR procedures to the standard Arnoldi process. In Section 3, based on the SOAR and MSOAR procedures, we propose the RSOAR method and discuss it in some detail. In Section 4, we establish upper bounds for the relative residual norms of approximating (refined) Ritz pairs. With them, we propose a reasonable criterion for deciding numerical deflation and breakdown. We then develop implicitly restarted RSOAR and MSOAR algorithms and discuss how to select reasonable shifts for each algorithm and propose the exact shifts and the refined shifts for the MSOAR and RSOAR algorithms, respectively. We present an effective approach to treat numerical deflations. In Section 5, we report numerical experiments to illustrate the efficiency of the restarted algorithms and the superiority of the refined algorithm.

Throughout the paper, we denote by  $\|\cdot\|$  the spectral norm of a matrix and the 2-norm of a vector, by  $\|\cdot\|_1$  the 1-norm of a matrix, by  $I$  the identity matrix with the order clear from the context, by the superscripts  $T$  and  $*$  the transpose and conjugate transpose of a matrix, by  $\mathcal{C}^k$  the complex space of dimension  $k$  and by  $\mathcal{R}^{(k+1) \times k}$  the set of  $(k+1) \times k$  real matrices. We denote by  $\sigma_{\min}(F)$  the smallest singular value of a matrix  $F$  and by the Matlab notation  $A(i:j, k:l)$  the submatrix consisting of columns  $i$  to  $j$  and rows  $k$  to  $l$  of  $A$ .

## 2 Second-order Krylov subspaces, the SOAR procedure, the MSOAR procedure and the SOAR and MSOAR methods

Bai and Su [3] introduce the following concepts.

**Definition 1.** Let  $A$ ,  $B$  be matrices of order  $n$  and the vector  $u \neq 0$ , and define

$$\begin{aligned} r_0 &= u, \\ r_1 &= Ar_0, \\ r_j &= Ar_{j-1} + Br_{j-2} \quad \text{for } j \geq 2. \end{aligned}$$

Then  $r_0, r_1, r_2, \dots, r_{k-1}$  is called a second-order Krylov sequence based on  $A, B$  and  $u$  and  $\mathcal{G}_k(A, B; u) = \text{span}\{r_0, r_1, r_2, \dots, r_{k-1}\}$  a  $k$ th second-order Krylov subspace.

Note that (3) is a linearization of (4). Define the matrix

$$H = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \quad (5)$$

of order  $2n$ . For a  $2n$  dimensional starting vector  $v$ , we can generate a Krylov subspace  $\mathcal{K}_k(H, v) = \text{span}\{v, Hv, H^2v, \dots, H^{k-1}v\}$ . Particularly, if we choose  $v = [u^T, 0]^T$ , we have

$$\begin{bmatrix} r_j \\ r_{j-1} \end{bmatrix} = H^j v, \quad j \geq 0 \text{ with } r_{-1} = 0. \quad (6)$$

This indicates that the upper half part of  $\mathcal{K}_k(H, v)$  is just  $\mathcal{G}_k(A, B; u)$  and its lower half part is just  $\mathcal{G}_{k-1}(A, B; u)$ , respectively. These fundamental facts can be compactly expressed as

$$\mathcal{G}_{k-1}^2(A, B; u) \subset \mathcal{K}_k(H, v) \subset \mathcal{G}_k^2(A, B; u), \quad (7)$$

where  $\mathcal{G}_k^2(A, B; u)$  is the subspace generated by

$$\left\{ \begin{bmatrix} r_0 \\ 0 \end{bmatrix}, \begin{bmatrix} r_1 \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} r_{k-1} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ r_0 \end{bmatrix}, \begin{bmatrix} 0 \\ r_1 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ r_{k-1} \end{bmatrix} \right\};$$

see also Theorem 2.4 in [22]. Due to the equivalence of QEP (4) and the eigenproblem of  $H$ , if the eigenvector  $[\lambda x^T, x^T]^T$  is contained in  $\mathcal{K}_k(H, [u^T, 0]^T)$ , then the eigenvector  $x$  of QEP (4) is contained in  $\mathcal{G}_k(A, B; u)$ . Therefore, continuity arguments suggest that if there is a good approximation on  $x$  contained in  $\mathcal{G}_k(A, B; u)$  then there is an approximation with comparable accuracy on  $[\lambda x^T, x^T]^T$  in  $\mathcal{K}_k(H, [u^T, 0]^T)$ . These illustrate that for QEP (4) the subspace  $\mathcal{G}_k(A, B; u)$  provides the same information as  $\mathcal{K}_k(H, v)$  for the eigenvalue problem of  $H$ . This motivates us to directly solve QEP (1) based on  $\mathcal{G}_k(A, B; u)$  rather than solve the eigenproblem of  $H$  based on  $\mathcal{K}_k(H, v)$ .

Bai and Su [3] propose the following procedure for computing an orthonormal basis  $\{q_j\}_{j=1}^k$  of  $\mathcal{G}_k(A, B; u)$  and an auxiliary vector sequence  $\{p_j\}$ .

#### Algorithm 1. SOAR procedure

```

1:  $q_1 = u/\|u\|$ 
2:  $p_1 = 0$ 
3: for  $j = 1, 2, \dots, k$  do
4:    $r = Aq_j + Bp_j$ 
5:    $s = q_j$ 
6:   for  $i = 1, 2, \dots, j$  do
7:      $t_{ij} = q_i^* r$ 
8:      $r = r - q_i t_{ij}$ 
9:      $s = s - p_i t_{ij}$ 
10:  end for
11:   $t_{j+1j} = \|r\|$ 
12:  if  $t_{j+1j} = 0$ , stop
13:   $q_{j+1} = r/t_{j+1j}$ 
14:   $p_{j+1} = s/t_{j+1j}$ 
15: end for
```

This algorithm leads to the following basic results [3].

**Theorem 1.** Define  $Q_k = [q_1, q_2, \dots, q_k]$  and  $P_k = [p_1, p_2, \dots, p_k]$  and  $\hat{T}_k = \begin{bmatrix} T_k \\ t_{k+1k} e_k^* \end{bmatrix} = [t_{ij}] \in \mathcal{R}^{(k+1) \times k}$ . Then we have

$$\text{span}\{Q_k\} = \mathcal{G}_k(A, B; u) \quad (8)$$

and the SOAR decomposition

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} Q_k \\ P_k \end{bmatrix} = \begin{bmatrix} Q_{k+1} \\ P_{k+1} \end{bmatrix} \hat{T}_k, \quad (9)$$

where  $Q_{k+1} = [Q_k, q_{k+1}]$ ,  $P_{k+1} = [P_k, p_{k+1}]$ .

(9) is an Arnoldi like decomposition on  $H$  starting with  $v = [q_1^T, 0]^T$ , but the columns of

$$\begin{bmatrix} Q_k \\ P_k \end{bmatrix}$$

are a non-orthogonal basis of  $\mathcal{K}_k(H, v)$ . Furthermore,  $q_1, q_2, \dots, q_k$  form an orthonormal basis of  $\mathcal{G}_k(A, B; u)$  and  $p_2, \dots, p_k$  form a non-orthogonal basis of the subspace  $\mathcal{G}_{k-1}(A, B; u)$ .

If Algorithm 1 stops prematurely at step  $j$ , there are two possibilities. One possibility is that  $r_i, i = 0, \dots, j$  are linearly dependent but  $[r_i^T, r_{i-1}^T]^T, i = 0, \dots, j$  with  $r_{-1} = 0$  are not. In this case,  $\mathcal{G}_{j+1}(A, B; u) = \mathcal{G}_j(A, B; u)$  but  $\mathcal{K}_{j+1}(H, v) \neq \mathcal{K}_j(H, v)$ , so the Arnoldi process on  $H$  does not terminate at step  $j$ ; see Lemma 2.4 of [22]. This situation is called deflation. The other possibility is that both vector sequences  $\{r_i\}$  and  $\{[r_i^T, r_{i-1}^T]^T\}$  are linearly dependent at step  $j$ . This situation is called breakdown and the Arnoldi process on  $H$  terminates; see Theorem 3.1 of [3]. If deflation occurs in the SOAR procedure at step  $j$ ,  $\mathcal{K}_j(H, v)$  does not contain any eigenvector of  $H$ , which implies that  $\mathcal{G}_j(A, B; u)$  does not contain any eigenvector of QEP (4); if breakdown occurs in the SOAR procedure at step  $j$ ,  $\mathcal{G}_j(A, B; u)$  contains  $2j$  exact eigenvectors of QEP (4). Therefore, deflation must be remedied.

Bai and Su [3] present the following algorithm that can remedy deflation.

**Algorithm 2. SOAR procedure with deflation**

```

1:  $q_1 = u/\|u\|$ 
2:  $p_1 = 0$ 
3: for  $j = 1, 2, \dots, k$  do
4:    $r = Aq_j + Bp_j$ 
5:    $s = q_j$ 
6:   for  $i = 1, 2, \dots, j$  do
7:      $t_{ij} = q_i^* r$ 
8:      $r = r - q_i t_{ij}$ 
9:      $s = s - p_i t_{ij}$ 
10:  end for
11:   $t_{j+1j} = \|r\|$ 
12:  if  $t_{j+1j} = 0$ 
13:    if  $s \in \text{span}\{p_i | i : q_i = 0, 1 \leq i \leq j\}$ 
14:      break
15:    else deflation
16:      reset  $t_{j+1j} = 1$ 
17:       $q_{j+1} = 0$ 
18:       $p_{j+1} = s$ 
19:    end if
20:  else
21:     $q_{j+1} = r/t_{j+1j}$ 
22:     $p_{j+1} = s/t_{j+1j}$ 
23:  end if
24: end for

```



In the above procedure, if  $t_{j+1j} = 0$  at line 12 and deflation occurs, we simply take  $q_{j+1} = 0$  and set  $t_{j+1j}$  to one. To decide if  $s \in \text{span}\{p_i | i : q_i = 0, 1 \leq i \leq j\}$ , the Gram–Schmidt orthogonalization with iterative refinement can be used, as suggested in [3, 22]. The procedure can continue until it breaks down. When deflation occurs, the nonzero vectors in the sequence  $\{q_i\}$  still span the second-order Krylov subspace  $\mathcal{G}_k(A, B; u)$  whose dimension is smaller than  $k$ . For details on Algorithm 2, we refer the reader to Bai and Su [3], where they present a SOAR procedure with deflation and memory savings by exploiting  $p_1 = 0$ .

For the SOAR procedure with deflation, Theorem 1 and relation (9) are still true. Because of (7), we can easily justify the following result.

**Theorem 2.** *If the Arnoldi process on  $H$  breaks down at step  $k$ , Algorithm 2 breaks down at step  $k - 1$ ; if Algorithm 2 breaks down at step  $k$ , the Arnoldi process breaks down at step  $k$ .*

As was realized [21, 22], a serious disadvantage of the SOAR procedure is that the implicit restarting technique cannot be applied to it since the updated  $p_1$  is not zero any more when implicitly restarting it. Otto [22] proposed modifying the original starting vector  $p_1 = 0$  in Algorithms 1–2 and allowing it to be nonzero. This leads to a modified second-order Krylov subspace and a modified second-order Arnoldi (MSOAR) procedure that generates an orthonormal basis of it, as shown below.

**Definition 2.** [22] *Let  $A$  and  $B$  be  $n \times n$  matrices and  $u_1, u_2 \in \mathbb{C}^n$  nonzero vectors, and define the sequence*

$$\begin{aligned} r_0 &= u_1, \\ r_1 &= Ar_0 + Bu_2, \\ r_j &= Ar_{j-1} + Br_{j-2} \quad \text{for } j \geq 2. \end{aligned}$$

*Then  $r_0, r_1, r_2, \dots, r_{k-1}, u_2$  is called the  $k$ th second-order Krylov sequence based on  $A, B$  and  $u_1, u_2$ , and  $\mathcal{G}_k(A, B; u_1, u_2) = \text{span}\{r_0, r_1, r_2, \dots, r_{k-1}, u_2\}$  the  $k$ th second-order Krylov subspace.*

Note that the subspace  $\mathcal{G}_k(A, B; u_1, u_2)$  is spanned by the upper block of the sequence

$$\left\{ \begin{bmatrix} r_0 \\ u_2 \end{bmatrix}, \begin{bmatrix} r_1 = Ar_0 + Bu_2 \\ r_0 \end{bmatrix}, \dots, \begin{bmatrix} r_{k-1} = Ar_{k-2} + Br_{k-3} \\ r_{k-2} \end{bmatrix}, \begin{bmatrix} u_2 \\ 0 \end{bmatrix} \right\}. \quad (10)$$

It is clear that the subspace spanned by its upper block vectors contains the subspace spanned by its lower block vectors. For  $\mathcal{K}_k(H, \tilde{v})$  with  $\tilde{v} = [u_1^T, u_2^T]^T$ , we easily see its upper and lower half parts are contained in  $\mathcal{G}_k(A, B; u_1, u_2)$  and  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ , respectively, and vice versa. Similar to (7), these simple but fundamental facts can be expressed as

$$\mathcal{G}_{k-1}^2(A, B; u_1, u_2) \subset \mathcal{K}_k(H, \tilde{v}) \subset \mathcal{G}_k^2(A, B; u_1, u_2). \quad (11)$$

Therefore, for QEP (4),  $\mathcal{G}_k(A, B; u_1, u_2)$  provides the same information as  $\mathcal{K}_k(H, \tilde{v})$  for the eigenproblem of  $H$ .

Algorithm 3 describes the MSOAR procedure [22] that can remedy deflation and generates an orthonormal basis of  $\mathcal{G}_k(A, B; u_1, u_2)$  of dimension  $k + 1$ .

**Algorithm 3. MSOAR procedure**

**Input:** Matrices  $A, B$ , nonzero vectors  $u_1, u_2$ , and steps  $k$ .

**Output:** If no premature stop occurs, the nonzero vectors among  $q_1, \dots, q_{k+1}$  form an orthonormal basis for  $\mathcal{G}_k(A, B; u_1, u_2)$ , the sequence  $\{p_1, \dots, p_k\}$  spans  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$  and a  $(k + 1) \times k$  upper Hessenberg matrix  $\hat{T}_k = [t_{ij}]$  is returned.

**1:**  $q_1 = \frac{u_1}{\|u_1\|}$ ,  $p_1 = \frac{u_2}{\|u_2\|}$ ,  $l = 0$ .

**2:** for  $j = 1, 2, \dots, k$  do

**3:**  $r = Aq_j + Bp_j$

```

4:    $s = q_j$ 
5:   for  $i = 1, 2, \dots, j$  do
6:      $t_{ij} = q_i^* r$ 
7:      $r = r - t_{ij} q_i$ 
8:      $s = s - t_{ij} p_i$ 
9:   end for
10:   $t_{j+1j} = \|r\|$ 
11:  if  $t_{j+1j} = 0$ 
12:     $l = l + 1, t_{j+1j} = 1, q_{j+1} = 0, p_{j+1} = s, f_l = p_{j+1}.$ 
13:    if  $(l = 0 \& p_{j+1} \in \text{span}\{q_1, \dots, q_j, p_1\})$  or  $(l > 0 \& p_{j+1} \in \text{span}\{f_1, \dots, f_{l-1}\})$ 
14:      break, go to step 21.
15:    end if
16:  else
17:     $q_{j+1} = r / t_{j+1j}$ 
18:     $p_{j+1} = s / t_{j+1j}$ 
19:  end if
20: end for
21:  $q_{j+2} = p_1$ 
22: for  $i = 1 : j + 1$  do
23:    $\gamma = q_i^* q_{j+2}$ 
24:    $q_{j+2} = q_{j+2} - \gamma q_i$ 
25: end for
26: if  $\|q_{j+2}\| \neq 0$ 
27:    $q_{j+2} = \frac{q_{j+2}}{\|q_{j+2}\|}$ 
28: end if

```

Like the SOAR procedure, the MSOAR procedure may stop prematurely at step  $j < k$ . If so, deflation or breakdown will take place. That is, if  $r_0, r_1, \dots, r_{j-1}, u_2$  are linearly dependent but  $\begin{bmatrix} r_0 \\ u_2 \end{bmatrix}, \begin{bmatrix} r_1 \\ r_0 \end{bmatrix}, \dots, \begin{bmatrix} r_{j-1} \\ r_{j-2} \end{bmatrix}, \begin{bmatrix} u_2 \\ 0 \end{bmatrix}$  are linearly independent, then deflation occurs; if both vector sequences are linearly dependent, then breakdown occurs.

From Algorithm 3 and the key relation (11), we can easily establish the following results, which are analogues of Theorems 1–2.

**Theorem 3.** *The columns of  $Q_{k+1}$  generated by the  $k$ -step MSOAR procedure form an orthonormal basis of  $\mathcal{G}_k(A, B; u_1, u_2)$ , (9) holds and the columns of  $\begin{bmatrix} Q_k \\ P_k \end{bmatrix}$  form a non-orthogonal basis of  $\mathcal{K}_k(H, \tilde{v})$ . Furthermore, if the Arnoldi process on  $H$  breaks down at step  $k$ , then Algorithm 3 breaks down at step  $k - 1$ ; if Algorithm 3 breaks down at step  $k$ , the Arnoldi process on  $H$  breaks down at step  $k$ .*

This equivalence theorem indicates that breakdown is a lucky event since there are  $2k$  exact eigenvectors of QEP (1) contained in  $\mathcal{G}_k(A, B; u_1, u_2)$  if Algorithm 3 breaks down at step  $k$  and no deflation occurs before step  $k$ . However, if deflation occurs, the situation becomes completely different, as the following results reveal.

**Theorem 4.** *Assume that  $l$  is the smallest integer such that*

$$\mathcal{G}_{l+1}(A, B; u_1, u_2) = \mathcal{G}_l(A, B; u_1, u_2).$$

*Then we must have*

$$\mathcal{K}_{l+1}(H, \tilde{v}) \neq \mathcal{K}_l(H, \tilde{v}).$$



*Proof.* From the definition of  $\mathcal{K}_l(H, \tilde{v})$ , the assertion amounts to proving that

$$\gamma_0 \begin{bmatrix} r_0 \\ u_2 \end{bmatrix} + \gamma_1 \begin{bmatrix} r_1 \\ r_0 \end{bmatrix} + \dots + \gamma_l \begin{bmatrix} r_l \\ r_{l-1} \end{bmatrix} = 0 \quad (12)$$

holds if and only if  $\gamma_0 = \gamma_1 = \dots = \gamma_l = 0$ . Under the assumption on  $l$ , we know  $r_0, \dots, r_l, u_2$  are linearly dependent but  $r_0, \dots, r_{l-1}, u_2$  are linearly independent. So

$$\sum_{j=0}^{l-1} \alpha_j r_j + \alpha_l u_2 = 0 \text{ means that } \alpha_0 = \alpha_1 = \dots = \alpha_l = 0. \quad (13)$$

Since (12) can be written as

$$\begin{aligned} \gamma_0 r_0 + \gamma_1 r_1 + \dots + \gamma_l r_l &= 0, \\ \gamma_0 u_2 + \gamma_1 r_0 + \dots + \gamma_l r_{l-1} &= 0, \end{aligned}$$

from (13) we must have  $\gamma_0 = \dots = \gamma_l = 0$ . Therefore, the assertion is proved.  $\square$

An important consequence of this theorem is that the first premature stop of Algorithm 3 must be deflation rather than breakdown and no exact eigenvector of QEP (1) is contained in  $\mathcal{G}_l(A, B; u_1, u_2)$ , so we must continue the algorithm for more steps.

The SOAR and MSOAR methods are orthogonal projection methods. They project (1) onto  $\mathcal{G}_k(A, B; u)$  and  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ , respectively, and solves a  $k \times k$  projected QEP

$$(\theta^2 M_k + \theta C_k + K_k)g = 0, \quad (14)$$

where  $M_k = Q_k^* M Q_k$ ,  $C_k = Q_k^* C Q_k$  and  $K_k = Q_k^* M Q_k$ . It is clear that (14) preserves essential structure of the original QEP (1), such as possible symmetries, skew-symmetries, positiveness of  $M, C$  and  $K$  as well as possible spectrum symmetries. Suppose that the  $(\theta, g)$  are the eigenpairs of (14). Then the SOAR or MSOAR method uses the Ritz pairs  $(\theta, y = Q_k g)$  to approximate some of the eigenpairs of (1). The  $\theta$  and  $y$  are called the Ritz values and Ritz vectors of (1) with respect to  $\mathcal{G}_k(A, B; u)$  or  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ . If Algorithm 3 breaks down at step  $k$  and no deflation occurs before step  $k$ , the MSOAR method will find at least  $k$  exact eigenpairs of (1), as shown in [9].

### 3 A refined second-order Arnoldi (RSOAR) method

As is well known, for a sequence of subspaces containing increasingly accurate approximations to the desired eigenvectors, orthogonal projection methods may fail to compute eigenvectors [10, 19]. To correct this deficiency, a refined projection principle is proposed in [11, 13] (see also [25, 28]) for the linear eigenvalue problem that extracts the best approximate eigenvectors from a given subspace in the sense that the residuals formed with certain approximate eigenvalues available are minimized over the subspace. The refined projection methods correct possible non-convergence of eigenvectors and computationally viable. The refined projection principle has been extended to solve the SVD problems [8, 17, 18, 20]. Since the SOAR and MSOAR methods are orthogonal projection (Rayleigh–Ritz) methods, they also have the possible non-convergence for computing eigenvectors, as has already been shown [9]. To this end, the refined projection principle is extended to the QEP that finds a certain refined Ritz vector whose convergence is guaranteed when the projection subspace is accurate enough [9]. Below we combine the SOAR procedure with the refined projection principle and propose a refined SOAR or MSOAR method.

Suppose that we have computed an approximate eigenvalue  $\theta$  by the SOAR or MSOAR method. The Refined SOAR or MSOAR method, written as the RSOAR method hereafter, seeks a unit length vector  $\tilde{u} \in \mathcal{G}_k(A, B; u)$  or  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$  satisfying the optimal requirement

$$\|(\theta^2 M + \theta C + K)\tilde{u}\| = \min_{\substack{u \in \mathcal{G}_k(A, B; u) \text{ or } \mathcal{G}_{k-1}(A, B; u_1, u_2) \\ \|u\| = 1}} \|(\theta^2 M + \theta C + K)u\| \quad (15)$$

and uses it as an approximate eigenvector, called the refined Ritz vector or more generally a refined eigenvector approximation. Since the columns of  $Q_k$  form an orthonormal basis of  $\mathcal{G}_k(A, B; u)$  or  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ , (15) amounts to seeking a unit length vector  $\tilde{z} \in \mathcal{C}^k$  such that  $\tilde{u} = Q_k \tilde{z}$  with  $\tilde{z}$  the solution of

$$\min_{\substack{z \in \mathcal{C}^k \\ \|z\| = 1}} \|(\theta^2 M + \theta C + K)Q_k z\|. \quad (16)$$

Obviously,  $\tilde{z}$  is the right singular vector of the matrix  $\theta^2 M Q_k + \theta C Q_k + K Q_k$  associated with  $\sigma_{\min}(\theta^2 M Q_k + \theta C Q_k + K Q_k)$ . However, the direct computation of the SVD may be expensive. Assume that the matrix is real and  $k \ll n$ . If  $\theta$  is real, the cost of Golub–Reinsch’s SVD algorithm is about  $4nk^2$  flops and that of Chan’s SVD algorithm is about  $2nk^2$  flops [6, p. 254]. Suppose we want to compute  $m$  eigenpairs with  $m < k$ . The total costs are  $4nmk^2$  and  $2nmk^2$  flops, respectively.

The first author in [15] has proposed a cross-product matrix based algorithm for computing the SVD of a matrix, which can be much more efficient than the standard SVD algorithms. Applying the algorithm to our case, we form the cross-product matrix

$$B_k = (\theta^2 M Q_k + \theta C Q_k + K Q_k)^* (\theta^2 M Q_k + \theta C Q_k + K Q_k),$$

which is the symmetric (Hermitian) (semi-)positive definite and whose smallest eigenpair is  $(\sigma_{\min}^2(\theta^2 M Q_k + \theta C Q_k + K Q_k), \tilde{z})$ . We then compute the eigensystem of  $B_k$  by the QR algorithm to get  $\tilde{z}$ . In finite precision arithmetic, it is proved in [15] that the computed eigenvector is an approximation to  $\tilde{z}$  with accuracy  $O(\epsilon_{\text{mach}})$  and the square root of the Rayleigh quotient of  $B_k$  with respect to the computed eigenvector is an approximation to  $\sigma_{\min}(\theta^2 M Q_k + \theta C Q_k + K Q_k)$  with accuracy  $O(\epsilon_{\text{mach}})$  provided that the second smallest singular value of  $\theta^2 M Q_k + \theta C Q_k + K Q_k$  is not close to the smallest one, where  $\epsilon_{\text{mach}}$  is the machine precision.

Let us now look at the computational cost of this algorithm. Define

$$W_1 = M Q_k, W_2 = C Q_k, W_3 = K Q_k,$$

which are available when forming the projected QEP and do not need extra cost. Then

$$\begin{aligned} B_k = & |\theta|^4 W_1^* W_1 + |\theta|^2 W_2^* W_2 + W_3^* W_3 + \theta \bar{\theta}^2 W_1^* W_2 + \bar{\theta} \theta^2 W_2^* W_1 \\ & + \bar{\theta}^2 W_1^* W_3 + \theta^2 W_3^* W_1 + \bar{\theta} W_2^* W_3 + \theta W_3^* W_2, \end{aligned} \quad (17)$$

where the bar denotes the complex conjugate of a scalar. Assume that  $W_1, W_2$  and  $W_3$  are real and note that  $B_k$  is Hermitian for a complex  $\theta$  and real symmetric for a real  $\theta$ . Then we only need to form the upper (lower) triangular part of  $B_k$ , which involves the upper (lower) triangular parts of the nine matrices  $W_i^* W_j$ ,  $i, j = 1, 2, 3$ . So the total flops are about  $9nk^2$ . With  $W_i^* W_j$ ,  $i, j = 1, 2, 3$ , we only need  $O(k^3)$  flops to form  $B_k$  for either a real or complex  $\theta$ , negligible to  $9nk^2$  flops. So we only need  $9nk^2$  flops to form  $m$  Hermitian matrices  $B_k$ ’s for  $m$  approximate eigenvalues  $\theta$ ’s. We then compute the complete eigensystems of these  $m$

$B_k$ 's by the QR algorithm using  $O(mk^3)$  flops. This means that we can compute  $m$  right singular vectors  $\tilde{z}$ 's using about  $9nk^2$  flops when  $mk \ll n$ , a natural requirement in practice. In view of the above, we see that this cross-product based algorithm is more efficient than Golub–Reinsch's SVD algorithm when  $m \geq 3$  and Chan's SVD algorithm when  $m \geq 5$ .

We can now present a basic (non-restarted) RSOAR method.

**Algorithm 4. The RSOAR algorithm**

1. Given the starting vector  $u$  or  $u_1, u_2$ , run the SOAR or MSOAR procedure to generate an orthonormal basis  $Q_k$  of  $\mathcal{G}_k(A, B; u)$  or  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ .
2. Compute  $W_1 = MQ_k$ ,  $W_2 = CQ_k$  and  $W_3 = KQ_k$ .
3. Compute  $M_k = Q_k^*W_1$ ,  $C_k = Q_k^*W_2$  and  $K_k = Q_k^*W_3$ , solve the projected QEP

$$(\theta_i^2 M_k + \theta_i C_k + K_k)g_i = 0 \quad (18)$$

and select  $m$  Ritz values  $\theta_i$  as approximations to the  $m$  desired eigenvalues  $\lambda_i$ .

4. For each chosen  $\theta_i$ ,  $i = 1, 2, \dots, m$ , form  $B_k$  and compute the eigenvector  $\tilde{z}_i$  of  $B_k$  associated with its smallest eigenvalue by the cross-product matrix based algorithm.
5. Test accuracy of the  $m$  refined eigenpairs  $(\theta_i, \tilde{u}_i)$  by computing the relative residual norms

$$\frac{\|(\theta_i^2 M + \theta_i C + K)Q_k \tilde{z}_i\|}{\|M\|_1 + \|C\|_1 + \|K\|_1}, \quad (19)$$

where  $\|\cdot\|_1$  is the 1-norm.

## 4 Implicitly restarted algorithms

This section consists of four subsections. In Section 4.1, we consider how to determine deflation and breakdown numerically. We propose a reasonable criterion for deciding numerical deflation and breakdown. In Section 4.2, we briefly show why implicit restarting cannot be applied to the SOAR procedure and the MSOAR procedure is needed instead. Under the assumption that no deflation occurs, we review how to implicitly restart the MSOAR procedure and develop implicitly restarted MSOAR algorithm and RSOAR algorithm. In Section 4.3, we discuss how to select shifts and propose exact and refined shifts for two algorithms, respectively. In Section 4.4, we propose an effective approach to treat the deflation issue in implicit restarts, so that implicitly restarted algorithms can be run unconditionally and are of generality.

### 4.1 Determination of numerical deflation and breakdown

We consider the crucial issue of how to decide deflation and/or breakdown of Algorithm 3 numerically. In line 13, we use the Gram–Schmidt orthogonalization with refinement to determine if  $p_{j+1}$  is in  $\text{span}\{q_1, \dots, q_j, u_2\}$  or  $\text{span}\{f_1, \dots, f_{l-1}\}$ . In finite precision arithmetic,  $t_{j+1j}$  is rarely zero exactly and, in general, can only be numerically small. If  $t_{j+1j}$  satisfies

$$\frac{t_{j+1j}}{\|M\|_1 + \|C\|_1 + \|K\|_1} \leq \text{tol} \quad (20)$$

for some suitably small tolerance  $\text{tol}$ , we accept it as zero numerically. A reasonable choice for  $\text{tol}$  should satisfy the two requirements as stated in the introduction of Section 4:  $\text{tol}$  should

not be too small to cause the instability of Algorithm 3 since a very small  $tol$  may cause great growth in  $p_{j+1}$  in line 18;  $tol$  should make the MSOAR and RSOAR methods converge within a user-prescribed tolerance whenever numerical breakdown occurs. To meet these two requirements, we will prove that  $tol$  should be as small as the convergence tolerance.

If (20) is satisfied, numerical deflation and/or breakdown may take place. We decide whether or not it is numerical deflation in the following way: In line 13 of Algorithm 3, before the Gram–Schmidt orthogonalization is run, we first normalize  $f_1, \dots, f_{l-1}$  to have unit length. Let  $p$  be the resulting vector of orthogonalizing  $p_{j+1}$  against  $q_1, \dots, q_j, u_2$  or  $f_1, \dots, f_{l-1}$ . Then if  $\|p\| > tol$ , we accept it as numerical deflation; otherwise, numerical breakdown occurs.

Observe that the projected QEP (14) of the original large QEP (1) over  $span\{Q_k\}$  is equivalent to the generalized eigenvalue problem

$$\begin{bmatrix} -C_k & -K_k \\ I & 0 \end{bmatrix} \begin{pmatrix} \theta g \\ g \end{pmatrix} = \theta \begin{bmatrix} M_k & 0 \\ 0 & I \end{bmatrix} \begin{pmatrix} \theta g \\ g \end{pmatrix},$$

which is the projected problem of (2) over the subspace spanned by the columns of the orthonormal matrix

$$\hat{Q}_{2k} = \begin{bmatrix} \tilde{Q}_k & 0 \\ 0 & \tilde{Q}_k \end{bmatrix},$$

where  $\tilde{Q}_k$  is the matrix deleting zero column(s) of  $Q_k$ . The projected problem is further equivalent to the standard linear eigenvalue problem

$$\begin{bmatrix} -M_k^{-1}C_k & -M_k^{-1}K_k \\ I & 0 \end{bmatrix} \begin{pmatrix} \theta g \\ g \end{pmatrix} = \theta \begin{pmatrix} \theta g \\ g \end{pmatrix}.$$

Recall that the  $k$ -step MSOAR procedure can be written as

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} Q_k \\ P_k \end{bmatrix} = \begin{bmatrix} Q_k \\ P_k \end{bmatrix} T_k + t_{k+1k} \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} e_k^*,$$

$$q_{k+2} = \frac{(I - Q_{k+1}Q_{k+1}^*)p_1}{\|(I - Q_{k+1}Q_{k+1}^*)p_1\|},$$

with  $T_k = [t_{ij}]$  and  $e_k$  the  $k$ th coordinate vector of dimension  $k$ . Note that  $T_k$  is the oblique projection matrix of  $H = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}$  with the right subspace  $span\{[Q_k^T, P_k^T]^T\}$  and the left subspace  $span\{[Q_k^T, 0]^T\}$ . So the eigenvalues  $\nu_i$ ,  $i = 1, 2, \dots, k$  of  $T_k$  are the Petrov-Ritz values and the corresponding Petrov-Ritz vectors  $w_i = \begin{bmatrix} Q_k \\ P_k \end{bmatrix} s_i$  with  $s_i$  the normalized eigenvectors of  $T_k$  associated with the eigenvalues  $\nu_i$ .

Specially, if  $[Q_k^T, P_k^T]^T$  is column independent and the MSOAR procedure breaks down at step  $k$ , i.e.,  $t_{k+1k}[q_{k+1}^T, p_{k+1}^T]^T = 0$ , then  $span\{[Q_k^T, P_k^T]^T\}$  is an invariant subspace of  $H$ . It thus follows from Theorem 3 that the Arnoldi process on  $H$  starting with  $[q_1^T, p_1^T]^T$  breaks down at step  $k$  and the MSOAR or RSOAR method finds  $2\tilde{k}$  exact eigenpairs of (1), where  $\tilde{k}$  is the number of nonzero columns in  $\tilde{Q}_k$ .

Since  $span\{[Q_k^T, P_k^T]^T\} \subset span\{\hat{Q}_{2k}\}$ , the Ritz pairs  $(\theta, [\theta y^T, y^T]^T)$  of  $H$  over  $span\{\hat{Q}_{2k}\}$  are generally more accurate than those of  $H$  over  $span\{[Q_k^T, P_k^T]^T\}$  [10, 19] and thus more accurate than the above Petrov–Ritz pairs  $(\nu, w)$  [23, p. 141-2] and [28, p. 114-5]. We should comment that speaking of meaningful approximate eigenpairs only makes sense when  $span\{[Q_k^T, P_k^T]^T\}$  contains good approximations to the desired eigenvectors [10, 19, 25]. Based on these, we now make the following reasonable assumption.

**Assumption:**  $\text{span}\{[Q_k^T, P_k^T]^T\}$  is a good subspace and the Ritz pair  $(\theta, [\theta y^T, y^T]^T)$  is always at least as good as Petrov–Ritz pair  $(\nu, w)$  when approximating the same desired eigenpair  $(\lambda, x)$ , i.e., the residual norm of the former is no more than that of the latter.

**Theorem 5.** Assume that  $(\theta, [\theta y^T, y^T]^T)$  is more accurate than  $(\nu, w)$  in the sense of residual norm, and set

$$c_k = (|\theta|^2 + 1)^{1/2} \frac{(\|M\|^2 + \|p_{k+1}\|^2)^{1/2}}{(1 + \|P_k s\|^2)^{1/2}}. \quad (21)$$

Then for the MSOAR and RSOAR methods we have

$$\|(\theta^2 M + \theta C + K)y\| \leq c_k t_{k+1k} |e_k^* s|, \quad (22)$$

$$\|(\theta^2 M + \theta C + K)\tilde{u}\| \leq c_k t_{k+1k} |e_k^* s|. \quad (23)$$

If numerical breakdown occurs at step  $k$  in the sense of (20), then for the MSOAR and RSOAR methods we have

$$\frac{\|(\theta^2 M + \theta C + K)y\|}{\|M\|_1 + \|C\|_1 + \|K\|_1} \leq c_k |e_k^* s| \cdot \text{tol}, \quad (24)$$

$$\frac{\|(\theta^2 M + \theta C + K)\tilde{u}\|}{\|M\|_1 + \|C\|_1 + \|K\|_1} \leq c_k |e_k^* s| \cdot \text{tol}. \quad (25)$$

*Proof.* From (21) and  $A = -M^{-1}C$ ,  $B = -M^{-1}K$ , the residual of  $(\nu, w)$  as an approximate eigenpair of (2) is

$$r_\nu = \begin{bmatrix} -C & -K \\ I & 0 \end{bmatrix} \frac{w}{\|w\|} - \nu \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \frac{w}{\|w\|} = t_{k+1k} \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} \frac{e_k^* s}{\|w\|}. \quad (26)$$

Here we normalize  $w$  by its norm  $\|w\|$ . By  $\|s\| = 1$  and the definition of  $w$ , we have

$$\|w\| = (\|Q_k s\|^2 + \|P_k s\|^2)^{1/2} = (1 + \|P_k s\|^2)^{1/2}.$$

Therefore,

$$\|r_\nu\| = t_{k+1k} |e_k^* s| \frac{(\|M q_{k+1}\|^2 + \|p_{k+1}\|^2)^{1/2}}{(1 + \|P_k s\|^2)^{1/2}}. \quad (27)$$

We now investigate the residual  $r_\theta$  of  $(\theta, (\theta y^T, y^T)^T)$  with  $y = \tilde{Q}_k g$  as an approximate eigenpair of (2) and seek the relationship between  $r_\theta$  and the residual of  $(\theta, y)$  as an approximate eigenpair of (1). We have

$$\begin{aligned} r_\theta &= \begin{bmatrix} -C & -K \\ I & 0 \end{bmatrix} \begin{bmatrix} \theta y \\ y \end{bmatrix} / (|\theta|^2 + 1)^{1/2} - \theta \begin{bmatrix} M & \\ & I \end{bmatrix} \begin{bmatrix} \theta y \\ y \end{bmatrix} / (|\theta|^2 + 1)^{1/2} \\ &= \begin{bmatrix} -(\theta^2 M + \theta C + K)y \\ 0 \end{bmatrix} / (|\theta|^2 + 1)^{1/2}. \end{aligned}$$

Here we normalize  $(\theta y^T, y^T)^T$  by its norm  $(|\theta|^2 + 1)^{1/2}$ . Therefore, it exactly holds that

$$\|(\theta^2 M \tilde{Q}_k + \theta C \tilde{Q}_k + K \tilde{Q}_k)g\| = \|(\theta^2 M + \theta C + K)y\| = (|\theta|^2 + 1)^{1/2} \|r_\theta\|. \quad (28)$$

Since  $(\theta, [\theta y^T, y^T]^T)$  is assumed to be more accurate than  $(\nu, w)$  in the sense of residual norm, i.e.,

$$\|r_\theta\| \leq \|r_\nu\|,$$

we get from (27), (28) and (21) that

$$\begin{aligned} \|(\theta^2 M + \theta C + K)y\| &\leq (|\theta|^2 + 1)^{1/2} t_{k+1k} |e_k^* s| \frac{(\|M\|^2 + \|p_{k+1}\|^2)^{1/2}}{(1 + \|P_k s\|^2)^{1/2}} \\ &= c_k t_{k+1k} |e_k^* s|, \end{aligned}$$

from which and (20) it follows that (24) holds.

For the RSOAR method, since the residual norm of the refined approximate eigenpair  $(\theta, \tilde{u})$  is always no more than that of the MSOAR method, (23) and (25) are direct.  $\square$

(22) and (23) are a-posteriori computable bounds for residual norms of  $(\theta, y)$  and  $(\theta, \tilde{u})$ . We should point out that the upper bounds in (22), (23), (24) and (25) may be conservative since  $\text{span}\{[Q_k^T, P_k^T]^T\} \subset \text{span}\{\hat{Q}_{2k}\}$  may make  $\|r_\theta\| < \|r_\nu\|$  considerably.

Now we look into  $c_k$  when numerical breakdown occurs at step  $k$ . In this case, observe from Algorithm 3 that since we set  $t_{k+1k} = 1$ , we have

$$p_{k+1} = q_k - \sum_{i=1}^k t_{ik} p_i$$

with  $t_{ik} = q_i^*(Aq_k + Bp_k)$ . Therefore, we obtain

$$\begin{aligned} \|p_{k+1}\| &\leq 1 + \sum_{i=1}^k |t_{ik}| \|p_i\| \\ &\leq 1 + \sum_{i=1}^k (\|A\| + \|B\| \|p_k\|) \|p_i\| \\ &= 1 + \|A\| \sum_{i=1}^k \|p_i\| + \|B\| \|p_k\| \sum_{i=1}^k \|p_i\|. \end{aligned} \tag{29}$$

Since  $\|s\| = 1$ , we can roughly estimate

$$\frac{(\|M\|^2 + \|p_{k+1}\|^2)^{1/2}}{(1 + \|P_k s\|^2)^{1/2}} \approx \frac{(\|M\|_1^2 + \|p_{k+1}\|^2)^{1/2}}{(1 + \frac{1}{k} \sum_{j=1}^k \|p_j\|^2)^{1/2}}, \tag{30}$$

which can be monitored during the MSOAR procedure without computing  $s$ . Note that  $\|p_j\|$ ,  $j = 1, 2, \dots, k+1$  are controlled and bounded in Algorithm 3. By definition (21) of  $c_k$ , combining (30) with (29), we conclude that  $c_k$  is bounded and furthermore is moderate when  $\|M\|$  is not large.

**Remark 1.** Once  $|e_k^* s|$  is considerably smaller than one,  $c_k \cdot \text{tol}$  is a considerable overestimate on the relative residual norms in (24) and (25). As a result, if Algorithm 3 breaks down at step  $k$  in the sense of (20), then (24) and (25) indicate that the MSOAR and RSOAR methods generally find  $2\tilde{k}$  eigenpairs of (1) at least with accuracy at the level of  $\text{tol}$ .

**Remark 2.** Suppose that the convergence tolerance is  $\text{ctol}$ . Then (20), (24) and (25) show that if we choose  $\text{tol}$  such that

$$c_k |e_k^* s| \cdot \text{tol} \leq \text{ctol} \tag{31}$$

then the relative residual norms of  $(\theta, y)$  and  $(\theta, \tilde{u})$  definitely drop below  $\text{ctol}$ . Since  $c_k$  is moderate and  $|e_k^* s| \leq 1$ , we propose taking

$$\text{tol} = \text{ctol} \tag{32}$$

for the numerical breakdown and deflation. Numerical experiments will demonstrate that this choice works very well. Furthermore, they will show that such a  $\text{tol}$  is conservative and a bigger  $\text{tol}$  also works equally well, say,  $\text{ctol} = 10^{-10}$  and  $\text{tol} = 10^{-8}$ . This is expected as we may have  $|e_k^* s| < 1$  considerably.



## 4.2 Implicit restarts

As  $k$  increases, the SOAR or MSOAR method and the RSOAR method become costly and impractical due to storage requirement and/or computational cost. So it is generally necessary to restart them for practical purposes. That is, for a given maximum  $k$ , if the methods do not converge yet, we select a new starting vector  $u^+$  or  $u_1^+, u_2^+$  based on the information available to construct a better subspace  $\mathcal{G}_k(A, B; u^+)$  or  $\mathcal{G}_k(A, B; u_1^+, u_2^+)$  that contains richer information on the eigenvectors  $x$  associated with the desired eigenvalues  $\lambda$  and compute new better approximate eigenpairs with respect to  $\mathcal{G}_k(A, B; u^+)$  or  $\mathcal{G}_k(A, B; u_1^+, u_2^+)$ . Proceed in such a way until the methods converge.

For the Krylov subspace  $\mathcal{K}_k(H, v)$  with  $v = [u^T, 0]^T$ , if we restarted the Arnoldi process implicitly, an updated starting vector  $v^+$  would have the form

$$\gamma v^+ = \psi(H)v$$

with  $\gamma$  a normalizing factor such that  $\|v^+\| = 1$  and  $\psi(z)$  a filter polynomial of a certain degree. Obviously, for a given polynomial  $\psi(z)$ ,  $v^+$  does not preserve the structure of  $v$  any more. As a result, the new Krylov subspace  $\mathcal{K}_k(H, v^+)$  will lose those close connections to the SOAR procedure presented in Section 2 and we cannot correspond  $\mathcal{K}_k(H, v^+)$  to a second-order Krylov subspace. Thus, an implicit restarting scheme for the Krylov subspace  $\mathcal{K}_k(H, v)$  cannot be adapted to the SOAR method and its refined version. This suggests that we turn to the methods based on the MSOAR procedure, so that we can make use of the fundamental relation (11) to implicitly restart the methods based on the MSOAR procedure.

As was seen previously, the MSOAR method and the RSOAR method will find the  $2k$  exact eigenpairs of QEP (1) if the MSOAR procedure breaks down at step  $k$  and no deflation occurs before step  $k$ . Recall the basic fact that the Arnoldi method and the refined Arnoldi method on  $H$  find the  $k$  exact eigenpairs of  $H$  if the Arnoldi process breaks down at step  $k$ . The implicit restarting technique updates starting vectors repeatedly to generate new Krylov subspaces that are expected to contain increasingly more accurate approximations to the desired  $m(< k)$  eigenvectors until the Arnoldi process approximately breaks down at step  $m$  and an  $m$ -dimensional approximate invariant subspace of  $H$  is found. According to Theorem 3 and (11), we will have an  $m$ -dimensional approximate invariant subspace  $\mathcal{G}_{m-1}(A, B; u_1, u_2)$  of QEP (1) when  $\mathcal{K}_m(H, \tilde{v})$  is an approximate invariant subspace of  $H$ . Fundamentally, Theorem 3 and (11) indicate that restarting the MSOAR procedure, i.e., updating  $\mathcal{G}_m(A, B; u_1, u_2)$ , is equivalent to restarting the Arnoldi process, i.e., updating  $\mathcal{K}_m(H, \tilde{v})$ .

Under the crucial assumption that no deflation occurs, Otto [22] has proposed an implicitly restarted MSOAR algorithm with exact second-order shifts suggested. There, the exact second-order shifts are among the  $2k - m$  unwanted eigenvalues of the projected QEP of QEP (1) onto  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ . Otto's definition means that there might be many ways of choosing the exact second-order shifts. However, he does not suggest a definite one and simply says that any one of the candidates can be used a shift. As will be discussed in Section 4.2, selecting reasonable shifts among the candidates is mathematically nontrivial, and we must do more in order to select 'correct' ones.

Given  $p$  shifts  $\mu_1, \mu_2, \dots, \mu_p$ , performing  $p$  implicit shifted QR iterations on  $T_k$  yields

$$(T_k - \mu_1 I) \cdots (T_k - \mu_p I) = V_k \hat{R},$$

where  $V_k$  is a  $k \times k$  orthogonal matrix and  $\hat{R}$  is upper triangular. Specifically,  $V_k$  has only  $p$  nonzero subdiagonals. The following results are presented in [22].

**Theorem 6.** *Given  $p$  shifts  $\mu_1, \dots, \mu_p$ , perform  $p$  steps of implicit shifted QR iterations on  $T_k$ . Let  $\psi(T_k) = V_k R_k$  with  $\psi(\mu) = \prod_{j=1}^p (\mu - \mu_j)$  and define  $Q_k^+ = Q_k V_k$  and  $T_k^+ = V_k^* T_k V_k$ .*

Assume that no deflation occurs in the  $k(=m+p)$ -step MSOAR decomposition (21). Then we have an updated  $m$ -step MSOAR decomposition

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} Q_m^+ \\ P_m^+ \end{bmatrix} = \begin{bmatrix} Q_m^+ \\ P_m^+ \end{bmatrix} T_m^+ + \tilde{t}_{m+1m}^+ \begin{bmatrix} q_{m+1}^+ \\ p_{m+1}^+ \end{bmatrix} e_m^* \quad (33)$$

starting with  $\begin{bmatrix} q_1^+ \\ p_1^+ \end{bmatrix}$ , where  $Q_m^+ = Q_k V_k(:, 1:m)$ ,  $P_m^+ = P_k V_k(:, 1:m)$ ,  $T_m^+ = T_k^+(1:m, 1:m)$  is upper Hessenberg and

$$\begin{aligned} \begin{bmatrix} q_{m+1}^+ \\ p_{m+1}^+ \end{bmatrix} &= \frac{1}{\tilde{t}_{m+1m}^+} f_m^+, \\ f_m^+ &= t_{m+1m}^+ \begin{bmatrix} q_{m+1}^+ \\ p_{m+1}^+ \end{bmatrix} + t_{k+1k} V_k(k, m) \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix}, \\ \tilde{t}_{m+1m}^+ &= \|f_m^+\|. \end{aligned}$$

Theorem 6 states that if there is no deflation in (21) then we have naturally obtained an  $m$ -step MSOAR procedure after  $p$  implicit shifted QR iterations are run on  $T_k$ . (33) is then extended to a  $k$ -step one from step  $m+1$  upwards in a standard way other than restarting it from scratch.

Similar to a basic result in [24], we can prove the following theorem, which shows what the updated starting vector of (33) is. The result guides us how to suitably select shifts for use within implicitly restarting MSOAR and RSOAR algorithms.

**Theorem 7.** *It holds that*

$$\begin{bmatrix} q_1^+ \\ p_1^+ \end{bmatrix} = \frac{1}{\tau} \psi(H) \begin{bmatrix} q_1 \\ p_1 \end{bmatrix}, \quad (34)$$

with  $\psi(\lambda) = \prod_{j=1}^p (\lambda - \mu_j)$  and  $\tau$  a normalizing factor.

### 4.3 Selection of shifts

In principle, we are free to use different shifts in the implicitly restarted MSOAR procedure described above. For linear eigenvalue problems and SVD problems, it has been shown in [12, 14] and [17, 18] that the better the shifts approximate some of the unwanted eigenvalues or singular values, the richer information on the desired eigenvectors or singular vectors the updated starting vector contains, so that the resulting Krylov subspaces contain more accurate approximations to the desired eigenvectors or singular vectors and the implicitly restarted algorithms are expected to converge faster. It follows from (34) and (11) that this theory works for the implicitly restarted MSOAR and RSOAR algorithms as well, that is, increasingly better standard Krylov subspaces lead to increasingly better modified second-order Krylov subspaces and vice versa. So we should choose shifts for each algorithm in the sense that they are best possible approximations to some of the unwanted eigenvalues of (1).

We solve the projected QEP (14) and select  $m$  Ritz values  $\theta_i$  as approximations to the desired eigenvalues. Then we may use the unwanted Ritz values as shifts, called exact second-order shifts in [22]. The problem is that we now have  $2k$  Ritz values. If we used all the unwanted Ritz values as shifts, this would mean that we apply  $2k - m > p$  shifts to the  $k \times k$  matrix  $T_k$ . However, for a  $k$ -step MSOAR procedure, the number of shifts in implicit restarts cannot exceed  $p$ . So we face the problem of how to select  $p \leq k - m$  reasonable shifts among the  $2k - m$  candidates.

As is known, a QEP may have two distinct eigenvalues that share the same eigenvector; see [27]. This means that, for QEP (14), if shift candidates and the  $m$  Ritz values used to

approximate the desired eigenvalues share the common eigenvector(s), restarting will filter out the information on the desired eigenvector(s). Therefore, we must abandon such shift candidates. Otherwise, simply selecting arbitrary  $p$  ones among  $2k - m$  candidates, as was done in [22], may cause fatal consequence and make implicitly restarted algorithms fail to work.

To avoid filtering out the information on the desired eigenvectors and meanwhile to dampen those components of the unwanted eigenvectors, we suggest a new approach to select reasonable shifts. We project QEP (1) onto the orthogonal complement of  $\text{span}\{y_1, \dots, y_m\}$  with respect to  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ , where  $y_1, \dots, y_m$  are the Ritz vectors approximating the desired eigenvectors. Then we obtain a projected QEP of dimension  $p$  and compute its  $2p$  eigenvalues. It is distinctive that these  $2p$  eigenvalues are now approximations to some of the unwanted eigenvalues of QEP (1) since the information on  $x_1, \dots, x_m$  has been filtered out from  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ . So we can use any  $p$  ones of these  $2p$  candidates as shifts. We also call them exact shifts, which are, however, different from the exact second-order shifts or exact shifts in Otto [22]. To be unique, we choose the  $p$  ones *farthest* from the Ritz values  $\theta_i, i = 1, 2, \dots, m$ . If we are interested in  $m$  eigenvalues nearest to a target  $\sigma$  and/or the associated eigenvectors, QEP (1) can be equivalently transformed to a shift-invert QEP; see the end of Section 4.3. In this case, we select the  $p$  Ritz values among  $2p$  candidates *farthest* from  $\sigma$  as shifts. Such selection of shifts is motivated by the idea [17, 18], where some of the shifts are taken to be unwanted Ritz values farthest from the wanted approximate singular values. It was argued there that this selection may better dampen those components of the unwanted singular vectors and amplify the components of the desired singular vectors.

Algorithm 4 computes the refined Ritz vectors  $\tilde{u}_i$ . Since they are generally more and can be much more accurate than the Ritz vectors  $y_i$  [9, 16, 19], it is possible to find better shifts for use within the implicitly restarted RSOAR algorithm. The first author [12, 14] has proposed certain refined shifts for the refined Arnoldi method and the refined harmonic Arnoldi method for linear eigenvalue problems. Later on, the same idea has been extended to the refined Lanczos and harmonic Lanczos bidiagonalization methods [17, 18]. It is shown that the refined shifts are generally better than the corresponding exact shifts and can be computed efficiently and reliably. In the same spirit, we next propose certain refined shifts for the RSOAR method.

Since the refined Ritz vectors  $\tilde{u}_i, i = 1, 2, \dots, m$  are more accurate than the corresponding  $y_i$ , the orthogonal complement of  $\text{span}\{\tilde{u}_1, \dots, \tilde{u}_m\}$  with respect to  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$  contains richer information on the unwanted eigenvectors. As a result, the eigenvalues of the projected QEP of QEP (1) onto this complement are more accurate approximate eigenvalues than those of the projected QEP of QEP (1) onto the orthogonal complement of  $\text{span}\{y_1, \dots, y_m\}$  with respect to  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ . Therefore, they are generally better shifts than the exact shifts. We use the same approach as above to select  $p$  ones among them as shifts, called the refined shifts, for use within the implicitly restarted RSOAR algorithm.

Now we show how to compute the exact and refined shifts efficiently and reliably. We take the refined shifts as an example. The computation of exact shifts is analogous. Recall  $\tilde{u}_i = Q_k \tilde{z}_i, i = 1, 2, \dots, m$  and write  $Z_m = [\tilde{z}_1, \dots, \tilde{z}_m]$ . We comment that if two columns  $\tilde{z}_i$  and  $\tilde{z}_{i+1}$  of  $Z_m$  are complex conjugate then we replace them by their normalized real and imaginary parts, respectively, so that the resulting  $Z_m$  is real. Make the QR decomposition

$$Z_m = [U_m, U_\perp] \begin{bmatrix} R_m \\ 0 \end{bmatrix},$$

where  $U_m$  and  $U_\perp$  are  $k \times m$  and  $k \times p$  column orthonormal matrices, respectively, and  $R_m$  is  $m \times m$  upper triangular. We use the standard Matlab function `qr.m` to compute the decomposition in experiments. This costs  $O(k^3)$  flops, negligible to the cost of the  $k$ -step

MISOAR procedure. Then  $Q_k U_\perp$  is an orthogonal basis of the orthogonal complement of  $\text{span}\{\tilde{u}_1, \dots, \tilde{u}_m\}$  with respect to  $\mathcal{G}_{k-1}(A, B; u_1, u_2)$ . It is easily seen that the projected QEP of the large (1) onto  $\text{span}\{Q_k U_\perp\}$  is just the projected QEP of the small QEP (18) onto  $\text{span}\{U_\perp\}$ . Therefore, the projected QEP onto the orthogonal complement can be efficiently formed using  $O(k^3)$  flops. We then compute its  $2p$  eigenvalues using  $O(p^3)$  flops and select  $p$  ones among them as the refined shifts. The whole cost of computing the refined shifts is  $O(k^3)$  flops. For the exact shifts, recall the Ritz vectors  $y_i = Q_k g_i$ ,  $i = 1, 2, \dots, m$ . Write  $G_m = [g_1, \dots, g_m]$  and replace  $Z_m$  by it. We then compute the exact shifts in the same way as above.

#### 4.4 Treatment of deflations in implicit restarts

Previously it was assumed that no deflation occurs in implicit restarts. If deflation occurs at some step(s) in the MISOAR procedure,  $Q_k$  must have zero column(s) and  $Q_m^+ = Q_k V_k(:, 1:m)$  is not column orthonormal any longer and the matrix  $Q_m^+$  is not column orthonormal any longer since  $Q_k$  has zero column(s), so that (33) is not an  $m$ -step MISOAR decomposition any more. As a result, implicit restarting fails and is not applicable. Because of this, Otto [22] had to assume that deflation does not occur during the  $k$ -step MISOAR procedure. This limits the applicability and generality of the algorithm. In what follows we present an effective approach to treat the deflation issue, so that implicit restarting works unconditionally.

Suppose that deflation occurs at steps  $m_1, m_2, \dots, m_j \leq k$ . Then the corresponding  $j$  columns of  $Q_k$  are zeros. Denote by  $\hat{Q}_k$  and  $\hat{V}_k$  the matrices by deleting the zero columns of  $Q_k$  and rows  $m_1, m_2, \dots, m_j$  of  $V_k$ , respectively. Then it holds that  $Q_k^+ = Q_k V_k = \hat{Q}_k \hat{V}_k$ , from which and (21) we get

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} \hat{Q}_k \hat{V}_k \\ P_k V_k \end{bmatrix} = \begin{bmatrix} \hat{Q}_k \hat{V}_k \\ P_k V_k \end{bmatrix} T_k^+ + t_{k+1k} \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} e_k^* V_k, \quad (35)$$

where  $T_k^+ = V_k^* T_k V_k$ . Keep in mind that  $\hat{Q}_k$  itself is column orthonormal.

Note that  $\hat{V}_k$  is a  $(k-j) \times k$  matrix and its row rank is  $k-j$ . For brevity, we temporarily assume that the first  $k-j$  columns of  $\hat{V}_k$  are linearly independent, i.e., the matrix consisting the first  $k-j$  columns of  $\hat{V}_k$  is nonsingular. Applying the Gram-Schmidt orthogonalization with refinement to the columns of  $\hat{V}_k$ , we compute

$$\hat{V}_k = U_k R_k = [U_{k-j}, 0] \begin{bmatrix} R_{k-j} & R_{12} \\ 0 & \hat{R}_j \end{bmatrix}, \quad (36)$$

a variant of the QR decomposition, where  $U_{k-j}$  is a  $(k-j) \times (k-j)$  orthogonal matrix and  $R_k$  is a  $k \times k$  nonsingular upper triangular matrix, and during the orthogonalization we set  $R_k(i, i) = 1$ ,  $i = k-j+1, \dots, k$ , that is,  $\hat{R}_j(i, i) = 1$ ,  $i = 1, 2, \dots, j$ . We remark that  $U_{k-j} R_{k-j}$  is the QR decomposition of the matrix consisting of the first  $k-j$  columns of  $\hat{V}_k$ . Then we can transform (35) to

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} \hat{Q}_k U_k \\ P_k V_k R_k^{-1} \end{bmatrix} = \begin{bmatrix} \hat{Q}_k U_k \\ P_k V_k R_k^{-1} \end{bmatrix} R_k T_k^+ R_k^{-1} + t_{k+1k} \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} e_k^* V_k R_k^{-1}. \quad (37)$$

Since  $R_k^{-1}$  is upper triangular,  $R_k T_k^+ R_k^{-1}$  is Hessenberg. Note that  $V_k$  has only  $p = k - m$  nonzero subdiagonals. Then the first possible nonzero entry of  $e_k^* V_k$  is in position  $m$  and

$$t_{k+1k} e_k^* V_k R_k^{-1} = (0, \dots, 0, \tilde{\beta}, b^T)$$

with  $\tilde{\beta} = t_{k+1k} V_k(k, m) / e_m^* R_k e_m$  in position  $m$ . Equating the first  $m$  columns on both hand sides of (37), we obtain

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} \tilde{Q}_m^+ \\ \tilde{P}_m^+ \end{bmatrix} = \begin{bmatrix} \tilde{Q}_m^+ \\ \tilde{P}_m^+ \end{bmatrix} \tilde{T}_m^+ + \beta_m^+ \begin{bmatrix} q_{m+1}^+ \\ p_{m+1}^+ \end{bmatrix} e_m^*, \quad (38)$$

where  $\tilde{Q}_m^+ = \hat{Q}_k U_k(:, 1 : m)$ ,  $\tilde{P}_m = P_k V_k(:, 1 : m) R_m^{-1}$  with  $R_m$  the  $m \times m$  leading principal matrix of  $R_k$ ,  $\tilde{T}_m^+$  the  $m \times m$  leading principal matrix of  $R_k T_k^+ R_k^{-1}$ , and  $\begin{bmatrix} q_{m+1}^+ \\ p_{m+1}^+ \end{bmatrix} = \frac{1}{\beta_m^+} f_m^+$  with  $f_m^+ = \begin{bmatrix} \hat{Q}_k U_k \\ P_k V_k R_k^{-1} \end{bmatrix} e_{m+1} \tilde{t}_{m+1m}^+ + \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} \tilde{\beta}$  and  $\beta_m^+ = \|f_m^+\|$ .

Since  $\hat{Q}_k$  is column orthonormal,  $\tilde{Q}_m^+$  is column orthonormal too when  $U_k(:, 1 : m)$  is so. The column orthonormality of  $U_k(:, 1 : m)$  is guaranteed whenever  $m \leq k - j$ , i.e.,  $j \leq k - m$ . This means that  $\tilde{Q}_m^+$  is column orthonormal and no deflation occurs in (38) provided that the number of deflations during the last cycle of MSOAR procedure does not exceed  $k - m$ . If  $j > k - m$ , there are  $m - (k - j)$  deflations in (38) and  $m - (k - j)$  zero columns among  $\tilde{Q}_m^+$ , which correspond to the zero columns of  $U(:, 1 : m)$ . In both cases, it is easily justified that  $(\tilde{Q}_m^+)^* q_{m+1}^+ = 0$ . This means that (38) is a truly  $m$ -step MSOAR procedure. So we have successfully cured deflations in implicit restarts and developed a robust implicit restarting scheme for the MSOAR procedure.

Finally, we discuss the case that  $\hat{V}_k$  has row rank  $k - j$  but its first  $k - j$  columns of  $\hat{V}_k$  are linearly dependent. In this case,  $U_k$  in (36) still has  $k - j$  orthonormal columns but they are not the first  $k - j$  ones, and the  $j$  diagonal entries ones of  $R_k$  in (36) are not in the right bottom corner of  $R_k$  any more. If  $U_k(:, 1 : m)$  has zero columns, deflation occurs in (38) since  $\tilde{Q}_m^+ = Q_k U_k(:, 1 : m)$  has zero columns, which correspond to the zero columns of  $U_k(:, 1 : m)$ .

Having done the above, we have finally developed the following Algorithm 5.

**Algorithm 5. The implicitly restarted MSOAR type algorithms**

1. Given starting vectors  $q_1$  and  $p_1$ , the number  $m$  of desired eigenpairs and the integer  $p$  such that  $k = m + p$ , run the  $k$ -step MSOAR procedure to generate  $Q_k$ .
2. Do until convergence
 

Project QEP (1) onto  $\text{span}\{Q_k\}$  to get QEP (14) and solve it by the MSOAR method or the RSOAR method, respectively.

Determine the convergence of  $m$  desired approximate eigenpairs  $(\theta_i, y_i)$  or  $m$  refined approximate eigenpairs  $(\theta_i, \tilde{u}_i)$ , respectively.
3. If not converged, compute the  $p$  exact shifts or refined shifts and implicitly restart the MSOAR method with the  $p$  exact shifts or the RSOAR method with the  $p$  refined shifts, respectively.
4. EndDo

Algorithm 5 includes two algorithms: the implicitly restarted MSOAR algorithm with the exact shifts (IMSOAR) and RSOAR algorithm with the refined shifts (IRSOAR). We determine the convergence of a Ritz pair  $(\theta, y(= Q_k g))$  by the relative residual norm

$$\frac{\|(\theta^2 M + \theta C + K) Q_k g\|}{\|M\|_1 + \|C\|_1 + \|K\|_1}.$$

For the convergence of a refined Ritz pair  $(\theta, \tilde{u}(= Q_k \tilde{z}))$ , we replace the above  $g$  by  $\tilde{z}$ .

If  $m$  eigenvalues closest to a target  $\sigma$  are desired, we can use the shift-invert transformation  $\rho = \frac{1}{\lambda - \sigma}$  with  $\det(Q(\sigma)) \neq 0$  to transform QEP (1) to

$$\hat{Q}(\rho)x = (\rho^2 \hat{M} + \rho \hat{C} + \hat{K})x = 0,$$

where  $\hat{M} = \sigma^2 M + \sigma C + K$ ,  $\hat{C} = C + 2\sigma M$ ,  $\hat{K} = M$ . Let  $\hat{A} = -\hat{M}^{-1}\hat{C}$  and  $\hat{B} = -\hat{M}^{-1}\hat{K}$ . In all the previous algorithms, we use such  $\hat{A}$  and  $\hat{B}$ . Let  $(\tilde{\rho}, y)$  be an approximate eigenpair (either a Ritz or refined Ritz pair) of  $\hat{Q}(\rho)x = 0$  and  $\hat{r} = \hat{Q}(\tilde{\rho})y$ . Then  $(\frac{1}{\tilde{\rho}} + \sigma, y)$  is the corresponding approximate eigenpair of  $Q(\lambda)x = (\lambda^2 M + \lambda C + K)x = 0$ . We easily derive

$$\begin{aligned} \hat{r}/\tilde{\rho}^2 &= (\hat{M} + \hat{C}/\tilde{\rho} + \hat{K}/\tilde{\rho}^2)y \\ &= (\sigma^2 M + \sigma C + K + (C + 2\sigma M)/\tilde{\rho} + M/\tilde{\rho}^2)y \\ &= ((\frac{1}{\tilde{\rho}} + \sigma)^2 M + (\frac{1}{\tilde{\rho}} + \sigma)C + K)y = Q(\frac{1}{\tilde{\rho}} + \sigma)y = \tilde{r}, \end{aligned} \quad (39)$$

from which it is direct to get  $\|\tilde{r}\|$  from  $\|\hat{r}\|$  without computing  $\tilde{r}$  explicitly.

## 5 Numerical experiments

In this section we report numerical examples to illustrate the performance of IMSOAR and IRSOAR and the superiority of IRSOAR to IMSOAR. All the experiments were run on an Intel(R) core(TM)2 with CPU 1.86GHz and 2GB RAM using Matlab 7.1 with  $\epsilon_{\text{mach}} = 2.22 \times 10^{-16}$  under a Window XP system. We list CPU timings in seconds of the main parts in the two algorithms. The abbreviations of main parts are shown in Table 1, where we also list the time of computing residuals of approximate eigenpairs, which may not be negligible when a large number of eigenpairs are required since we have to compute them directly for different  $\theta$ 's. Note that SMALL is the time of solving small QEP's plus that of computing refined Ritz vectors for IRSOAR.

Table 1: Abbreviations

TOTAL	Total CPU time
SOAR	The CPU time of MSOAR procedure
EXP	The CPU time of explicit projection
IMP	The CPU time of implicit restarting
SMALL	The CPU time of solving small (projected) QEP's
RES	The CPU time of computing residuals
Restarts	The number of restarts

For each example, we used the same starting vector generated randomly in a uniform distribution for both algorithms. We transformed the projected QEP (14) to the generalized eigenvalue problem

$$\begin{bmatrix} -C_k & -K_k \\ I & 0 \end{bmatrix} \begin{bmatrix} \theta g \\ g \end{bmatrix} = \theta \begin{bmatrix} M_k & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \theta g \\ g \end{bmatrix}$$

and solved it by the QZ algorithm. We can recover an eigenvector  $g$  of QEP (14) from either the first  $k$  components or the last  $k$  components of  $[\theta g^T, g^T]^T$ . From the backward error analysis [7], it is preferable to take the first  $k$  ones if  $|\theta| \geq 1$  and the last  $k$  ones if  $|\theta| < 1$ . We adopted this choice. In all the examples, the breakdown or deflation tolerance is  $dtol$  and the



convergence tolerance is  $ctol$ . Based on (32) and the comments followed, we took  $dtol = ctol$  or  $dtol = 100 \times ctol$  in experiments.

**Example 1.** Consider the free vibration of a string with clamped ends in a spatially inhomogeneous environment [29]. The equation characterizing the wave motion is described by

$$\begin{cases} u_{tt} + \epsilon a(x)u_t = \Delta u, & x \in [0, \pi], \epsilon > 0, \\ u(t, 0) = u(t, \pi) = 0. \end{cases}$$

Approximating  $u(x, t)$  by  $\sum_{k=1}^n q_k(t) \sin(k\pi x)$  and applying the Galerkin method leads to a second-order differential equation

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = 0,$$

where  $q(t) = [q_1(t), \dots, q_n(t)]^T$ . We then get the QEP

$$(\lambda^2 M + \lambda C + K)x = 0,$$

where  $M = (\pi/2)I_n$ ,  $K = (\pi/2)\text{diag}(j^2)$  and

$$C = (c_{ij}), \quad c_{i,j} = \left| \int_0^\pi \epsilon a(x) \sin(ix) \sin(jx) dx \right|.$$

We took  $n = 1200$ ,  $\epsilon = 0.6$ , and  $a(x) = x^2(\pi - x)^2 - 201$ . We set  $\sigma = 0.6 + 0.8i$  and were interested in 60 eigenvalues nearest to  $\sigma$  and the corresponding eigenvectors. By taking  $dtol=10^{-8}$ ,  $ctol=10^{-10}$ , we tested the algorithms for various  $k$  and  $p$  and limited the maximum restarts to 100. Table 2 reported the results obtained by IRSOAR and IMSOAR.

Table 2: Example 1,  $dtol=10^{-8}$ ,  $ctol=10^{-10}$ .

Algorithm	$k$	$p$	TOTAL	SOAR	EXP	IMP	SMALL	RES	Restarts
IRSOAR	76	5	195.17	16.87	98.22	6.31	62.96	8.08	16
IMSOAR	76	5	143.28	17.31	103.11	6.73	5.54	8.63	17
IRSOAR	76	10	118.98	17.52	57.25	3.97	34.05	4.86	9
IMSOAR	76	10	86.23	17.38	55.66	3.91	3.05	4.77	9
IRSOAR	76	15	107.95	20.52	49.77	3.84	28.3	4.14	8
IMSOAR	76	15	240.77	46.41	145.87	21.69	7.54	11.78	25
IRSOAR	86	16	170.55	22.83	75.75	5.98	47.55	5.31	9
IMSOAR	86	16	134.05	24.84	83.68	6.77	4.7	5.61	10
IRSOAR	96	16	165.66	21.97	72.06	5.80	55.16	4.53	7
IMSOAR	96	16	109.77	21.86	70.93	5.84	5.22	4.55	7

We see from the table that for fixed subspace dimension  $k$  IRSOAR with larger  $p$  used fewer restarts and CPU timings, as expected, but it may not be the case for IMSOAR. For  $k = 76$ ,  $p = 15$ , IMSOAR used 25 restarts, much more than those used with the smaller  $p$ . This is not unusual since SOAR may converge irregularly. More precisely, the residual norms obtained by MSOAR and its restarted versions may behave irregular even if projection subspaces are increasingly improved and contain more accurate approximations to the desired eigenvectors; see [9] for a theoretical justification. We found that from the fourteenth to the last restart IMSOAR with  $k = 76$ ,  $p = 15$  had one deflation at each restart and broke down at step 63 in the last restart, delivering an invariant subspace of dimension 62 and thus 62 converged eigenpairs whose norms were smaller than  $dtol$ . For  $k = 86$  and  $p = 16$ , IRSOAR had one deflation at each of the last two restarts and IMSOAR had one deflation at each of

the last four restarts. Both algorithms broke down at the very last restart. Since a large number of eigenpairs are desired, computing refined Ritz vectors was quite time consuming. As the table indicated, explicit projection and solving the projected QEP's dominated the whole computational cost of IRSOAR.

Figures 1–2 described convergence processes of two algorithms for  $k = 76, p = 15$  and  $k = 86, p = 16$ , in which the left figures depicted the largest relative residual norms of approximate eigenpairs, while the right figures exhibited the number of deflations at each restart.

It was seen that IRSOAR computed all the desired eigenvalues after eight restarts and there was no deflation. In contrast, IMSOAR used 25 restarts and one deflation occurred at each of the 14th to the 20th restarts. When we enlarged  $k$  to 86 and set  $p = 16$ , both algorithms converged quickly and there was no deflation during all restarts.

From the experiments, we observed that IMSOAR was sensitive to  $k$  and  $p$  but IRSOAR behaved very smooth and converged faster for larger  $k$  and fixed  $p$  as well as fixed  $k$  and larger  $p$ .

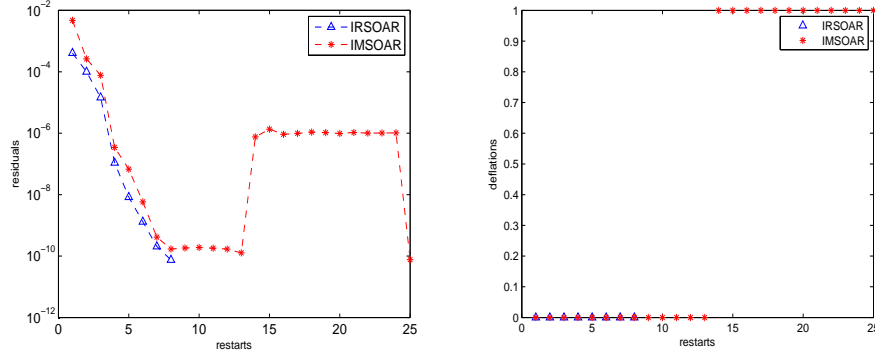


Figure 1: Example 1. Left: residuals versus restarts; right: deflations versus restarts.  $\sigma = 0.6 + 0.8i$ ,  $k = 76, p = 15$ ,  $dtol=10^{-8}$ ,  $ctol=10^{-10}$ .

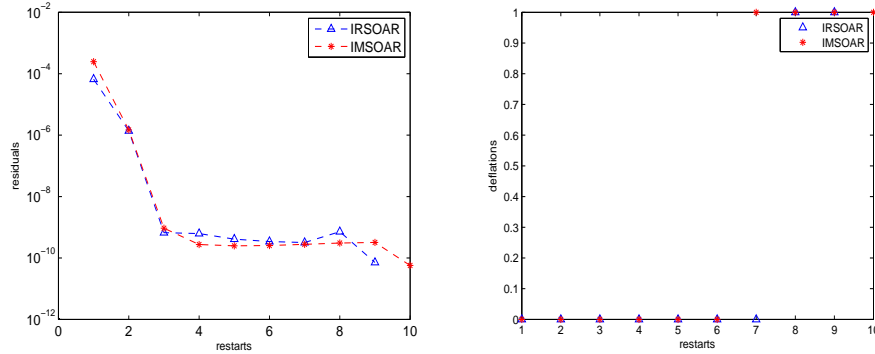


Figure 2: Example 1. Left: residuals versus restarts; right: deflations versus restarts.  $\sigma = 0.6 + 0.8i$ ,  $k = 86, p = 16$ ,  $dtol=10^{-8}$ ,  $ctol=10^{-10}$ .

We also tested two algorithms by computing six eigenvalues for  $dtol = ctol = 10^{-8}$  and various  $k$  and  $p$ . The pairs  $(k, p)$  were  $(20, 5)$ ,  $(20, 8)$ ,  $(20, 11)$ ,  $(25, 5)$  and  $(30, 5)$ . We found that both algorithms converged quickly and used no more than five restarts for all pairs. We

found that the restarts were reduced by using the same  $p$  and larger  $k$  for both algorithms. There was no deflation in both algorithms for all pairs.

**Example 2.** Consider the QEP arising from an  $n$ -degree-of-freedom damped mass-spring system [27]. By taking  $m_i = 1$  and letting all the springs (respectively, dampers) have the same constant  $\kappa$  (respectively,  $\tau$ ) except  $\kappa_1 = \kappa_n = 2\kappa$  and  $\tau_1 = \tau_n = 2\tau$ , the resulting matrices are

$$M = I, \quad C = \tau \cdot \text{tridiag}(-1, 3, -1), \quad K = \kappa \cdot \text{tridiag}(-1, 3, -1),$$

which are very sparse. We took  $n = 5000, \kappa = 5$  and  $\tau = 10$  and were interested in the six eigenvalues nearest to  $\sigma = -13 + 0.4i$  and the corresponding eigenvectors. For  $dtol = 10^{-8}$  and  $ctol = 10^{-10}$ , we tested IRSOAR and IMSOAR for some pairs  $(k, p)$  and limited the maximum restarts to 100. Table 3 lists the results, and Figures 3–4 depict the convergence processes for two sets of parameters  $(k, p)$ .

Table 3: Example 2,  $dtol=10^{-8}$ ,  $ctol=10^{-10}$ .

Algorithm	$k$	$p$	TOTAL	SOAR	EXP	IMP	SMALL	RES	Restarts
IRSOAR	40	15	96.52	24.24	9.14	27.17	18.12	7.02	57
IMSOAR	40	15	79.66	26.06	10.83	29.41	2.73	7.44	60
IRSOAR	40	23	97.23	31.75	9.49	26.20	17.00	6.44	55
IMSOAR	40	23	91.50	36.19	11.04	30.55	2.69	7.45	63
IRSOAR	40	31	125.86	46.78	11.56	33.81	20.26	8.11	65
IMSOAR	40	31	126.52	55.22	13.54	39.88	3.52	9.34	78
IRSOAR	45	31	107.31	39.61	9.50	28.50	17.80	5.80	44
IMSOAR	45	31	104.00	45.48	10.84	33.44	3.16	7.23	51
IRSOAR	50	31	87.05	26.36	8.11	24.75	15.88	4.66	32
IMSOAR	50	31	81.41	30.91	9.64	29.66	3.05	5.52	38

It can be found from Table 3 that IRSOAR used fewer restarts than IMSOAR for all given  $k$  and  $p$ . For fixed  $p = 31$ , restarts of both algorithms decreased with increasing  $k$ .

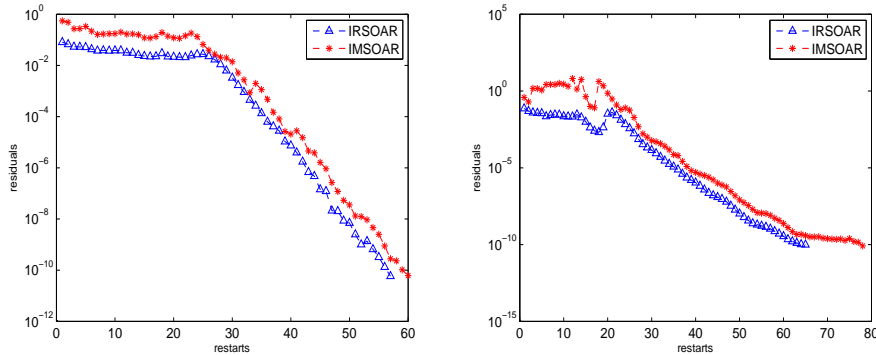


Figure 3: Example 2. Residuals versus restarts,  $dtol=10^{-8}$ ,  $ctol=10^{-10}$ . Left:  $k = 40, p = 15$ ; right:  $k = 40, p = 31$ .

Figures 3–4 indicates that for  $(k = 45, p = 31)$  IRSOAR and IMSOAR converged irregularly but for other  $(k, p)$  pairs the two algorithms converged quite smoothly. There was no deflation for both algorithms for all  $(k, p)$  pairs.

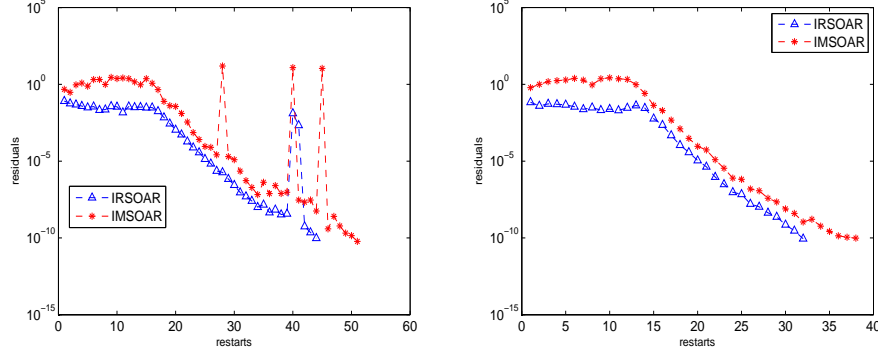


Figure 4: Example 2. Residuals versus restarts,  $dtol=10^{-8}$ ,  $ctol=10^{-10}$ . Left:  $k = 45$ ,  $p = 31$ ; right:  $k = 50$ ,  $p = 31$ .

**Example 3.** This problem comes from [2]. It is a nonlinear eigenvalue problem modeling a radio-frequency gun cavity that is of the form

$$T(\lambda)x = [K - \lambda M + i(\lambda - \sigma_1^2)^{1/2}W_1 + i(\lambda - \sigma_2^2)^{1/2}W_2]x = 0,$$

where  $M$ ,  $K$ ,  $W_1$ ,  $W_2$  are real symmetric matrices of size  $9956 \times 9956$ . From these matrices, we constructed the following QEP of the form

$$(\lambda^2 W_2 + \lambda M + K)x = 0.$$

We used IRSOAR and IMSOAR for  $dtol = 10^{-8}$ ,  $ctol = 10^{-8}$  to compute the 24 eigenvalues nearest to  $\sigma = 0.5 + 0.5i$  and the associated eigenvectors. We stopped the algorithms after 50 restarts were used. Table 4 reported the results.

Table 4: Example 3,  $dtol=10^{-8}$ ,  $ctol=10^{-8}$ .

Algorithm	$k$	$p$	TOTAL	SOAR	EXP	IMP	SMALL	RES	Restarts
IRSOAR	40	5	22.41	15.93	1.4	1.05	1.36	2.05	1
IMSOAR	40	5	—	—	—	—	—	—	50
IRSOAR	40	10	21.39	15.27	1.49	0.95	1.43	1.73	1
IMSOAR	40	10	—	—	—	—	—	—	50
IRSOAR	40	15	22.33	16.34	1.51	0.922	1.15	1.91	1
IMSOAR	40	15	—	—	—	—	—	—	50
IRSOAR	50	10	19.05	15.47	1.07	0	1.00	1.17	0
IMSOAR	50	10	—	—	—	—	—	—	50

It is remarkable to see that for  $k = 40$  and  $p = 5, 10$  and  $15$ , IRSOAR used only one restart while IMSOAR failed to converge after 50 restarts. So the IRSOAR is much more efficient than IMSOAR. We have also observed that there was no deflation for both algorithms. If we instead took a bigger  $k = 50$  and  $p = 10$ , we found that IRSOAR computed the desired eigenpairs without restart but IMSOAR still did not converge. So for this problem, IRSOAR worked very well but IMSOAR completely failed for given  $(k, p)$ .

To better look into the convergence behavior, Figure 5 depicted the convergence curves of both algorithms for  $k = 40$ ,  $p = 10$  and  $k = 40$ ,  $p = 15$ . We see that for IMSOAR the maximum relative residuals of Ritz pairs stayed roughly between  $10^{-6} \sim 10^{-4}$  and did not

decrease further as restarts proceeded. Furthermore, keep in mind that in the first cycle IRSOAR and IMSOAR used the same starting vector and computed approximate eigenpairs with respect to the same subspace. It is strikingly seen from Figure 5 that in the first cycle the maximum residual norms obtained by RSOAR improved those obtained by MSOAR by at least three orders, a very big gain. Experimentally, this confirms the theory of [9] and indicates that one can benefit much from the refined projection principle.

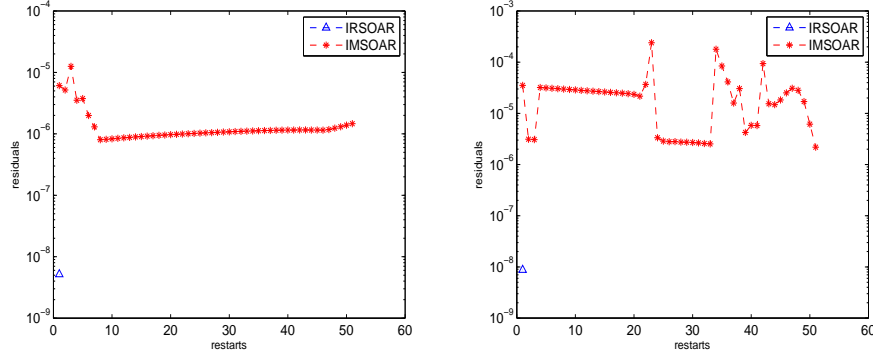


Figure 5: Example 3. Residuals versus restarts. Left:  $k = 40, p = 10$ ; right:  $k = 40, p = 15$ .

**Example 4.** In this example, the QEP  $Q(\lambda)x = (\lambda^2 M + \lambda C + (1 + i\mu)K)x = 0$  arises in a model of a concrete structure supporting a machine assembly [2, 5]. The matrices have dimension 2472, where  $M$  is real diagonal and of low rank, and  $C$  is the viscous damping matrix and is pure imaginary and diagonal. The factor  $1 + i\mu$  adds uniform hysteretic damping. The default is  $\mu = 0.04$ .

We ran IRSOAM and IMSOAR to compute the 30 eigenvalues of  $Q(\lambda)x = 0$  nearest to  $\sigma = 34 + 10i$  and the corresponding eigenvectors. Table 5 reported the results.

Table 5: Example 4,  $dtol=10^{-8}$ ,  $ctol=10^{-8}$ .

Algorithm	$k$	$p$	TOTAL	SOAR	EXP	IMP	SMALL	RES	Restarts
IRSOAR	42	5	19.13	6.48	1.16	2.25	4.31	2.48	9
IMSOAR	42	5	15.36	6.48	1.18	2.28	0.64	3.02	9
IRSOAR	42	8	13.97	6.26	0.72	1.34	2.79	1.13	5
IMSOAR	42	8	11.30	5.87	0.64	1.23	0.38	1.70	5
IRSOAR	42	11	10.94	5.88	0.54	1.02	1.45	1.75	4
IMSOAR	42	11	9.38	5.90	0.55	0.97	0.31	1.39	4
IRSOAR	45	8	11.77	5.62	0.68	1.17	2.22	1.23	4
IMSOAR	45	8	9.75	5.65	0.64	1.11	0.39	1.59	4
IRSOAR	50	8	11.78	5.41	0.66	1.05	2.48	1.50	3
IMSOAR	50	8	9.22	5.47	0.64	1.03	0.44	1.42	3

It is seen that this problem is easy to solve by both IRSOAR and IMSOAR. The algorithms converged smoothly and quickly. For fixed  $p = 8$  and  $k = 42, 45, 50$ , both algorithms used fewer restarts with increasing  $k$ . These phenomena are in accordance with the theory since subspaces of increasing dimensions  $k + p$  should generally contain more information on the desired eigenvectors. For  $p = 5$ , IRSOAR and IMSOAR had one deflation at the 39th, 40th and 41th restarts, and they broke down at the 41th restart. For  $p = 8$ , both algorithms had one deflation at each of the last four restarts, and broke down at the 42th restart.

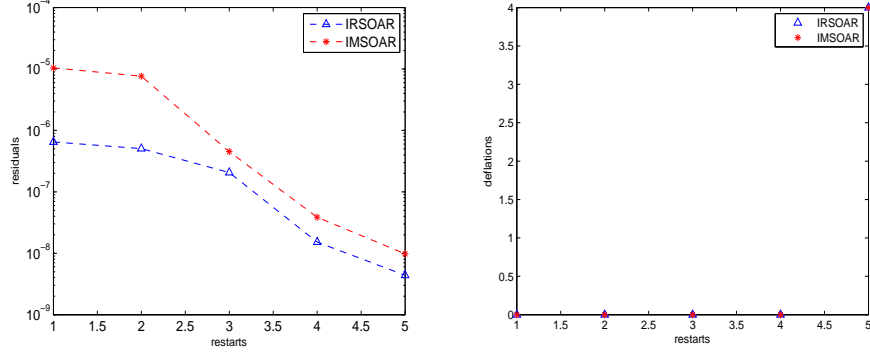


Figure 6: Example 4. Left: Residuals versus restarts; right: deflations versus restarts.  $k = 42, p = 8$ .

Figure 6 depicted the convergence processes and deflation details with  $k = 42, p = 8$ , from which we see that both IRSOAR and IMSOAR had four deflations at the last restart. In addition, the figure illustrates that although IRSOAR and IMSOAR both used five restarts to achieve the convergence, IRSOAR gave smaller residuals, i.e., more accurate approximate eigenpairs, at each restart.

**Example 5.** We consider the damped vibration mode of an acoustic fluid confined in a cavity with absorbing walls capable of dissipating acoustic energy [9]. We take the same geometrical data as in [9] and more properties on this problem can be found there. The QEP is

$$\lambda^2 M_u u + (\alpha + \lambda \beta) A_u + K_u u = 0,$$

where  $\alpha = 5 \times 10^4 N/m^3, \beta = 200 Ns/m^3$ . The dimension of this QEP is  $n = 46548$ .

We ran IRSOAM and IMSOAR to compute the 8 eigenvalues nearest to  $\sigma = 25 + 18i$  and the corresponding eigenvectors. We took  $dtol = 10^{-8}, ctol = 10^{-10}$  in experiments. Table 6 reported the results.

Table 6: Example 5,  $dtol=10^{-8}, ctol=10^{-10}$ .

Algorithm	$k$	$p$	TOTAL	SOAR	EXP	IMP	SMALL	RES	Restarts
IRSOAR	30	10	58.87	40.86	3.44	5.48	4.12	3.55	2
IMSOAR	30	10	146.06	91.25	10.19	24.13	0.25	10.98	9
IRSOAR	30	15	62.39	45.42	3.21	5.13	3.97	3.34	2
IMSOAR	30	15	86.20	60.98	4.60	10.44	0.13	5.09	4
IRSOAR	30	20	47.56	37.13	2.12	2.55	2.65	2.25	1
IMSOAR	30	20	385.64	288.99	19.22	46.89	0.38	20.66	18
IRSOAR	35	20	31.89	26.78	1.42	0	1.77	1.31	0
IMSOAR	35	20	30.15	26.66	1.46	0	0.05	1.34	0

The table clearly indicates that IRSOAR solved the problem very successfully and, regarding restarts, was several times (from nearly one to 17) faster than IMSOAR except for  $k = 35$ , in which case both algorithms found the desired eigenpairs without restart. We further observe that for fixed  $k$ , IRSOAR used fewer restarts with increasing  $p$  while IMSOAR converged quite irregularly with varying  $p$ : If  $p$  is increased from 10 to 15, the number of restarts was reduced from 9 to 4; if we changed  $p$  to 20, the number of restarts was increased



to 18. This may be due to the intrinsically irregular convergence behavior of the Rayleigh–Ritz method for QEP [9]. It is also seen that for fixed  $p = 20$  if we changed  $k$  from 30 to 35 then IMSOAR drastically converged without restart. These results demonstrate that, unlike IRSOAR, IMSOAR can be quite sensitive to parameter choices, so that the convergence behavior of IRSOAR is more pronounced than that of IMSOAR.

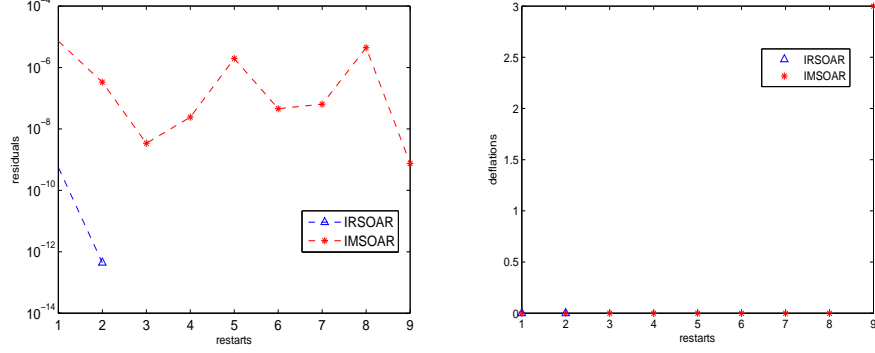


Figure 7: Example 5. Left: Residuals versus restarts; right: deflations versus restarts.  $k = 30, p = 10$ .

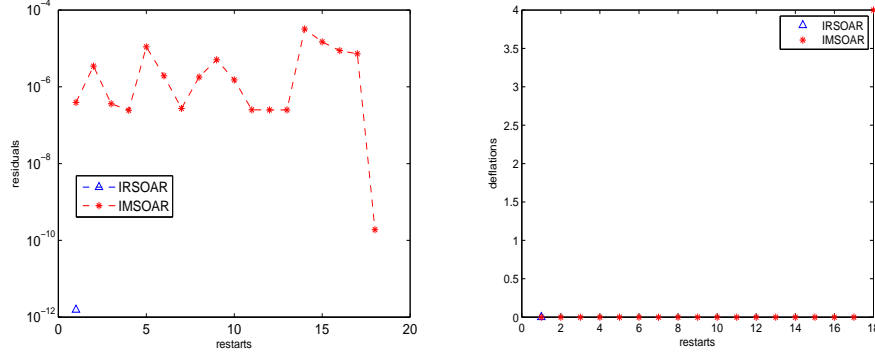


Figure 8: Example 5. Left: Residuals versus restarts; right: deflations versus restarts.  $k = 30, p = 20$ .

Figures 7–8 described the convergence processes of two algorithms for  $k = 30, p = 10$  and  $k = 30, p = 20$ . Obviously, IRSOAR converged smoothly and quickly and there was no deflation occurred. In contrast, IMSOAR converged irregularly for all given pairs. Furthermore, we observed that for  $k = 30$  and  $p = 10$ , IMSOAR had three deflations at steps 23, 25 and 26 and it broke down at the 26th step in the last restart; for  $k = 30, p = 20$ , IMSOAR had four deflations that occurred at steps 12, 13, 14 and 15 and broke down at the 15th step at the last restart, delivering 15 converged eigenpairs.

## 6 Conclusion

Based on the refined projection principle and the SOAR procedure, we have proposed a refined second-order Arnoldi (RSOAR) method that has the same structure preserving properties as the SOAR method. We have shown that the original SOAR method cannot be restarted

effectively and instead the MSOAR procedure is needed. We have established upper bounds for residual norms obtained by the MSOAR and RSOAR algorithms. From them, we have proposed reasonable criteria on numerical deflation and breakdown. The criteria ensure that the algorithms converge within the prescribed accuracy and meanwhile maintain the numerical stability of MSOAR procedure. For practical purposes, we have extended the implicit restarting technique to the MSOAR and RSOAR methods, developed their implicitly restarted algorithms IRSOAR and IMSOAR and proposed best possible shifts for each of them. We have shown how to compute the shifts efficiently and reliably.

Our final contribution is that, in order to overcome possible deflation and make implicit restarting unconditionally applicable, we have proposed an effective approach to handle the deflation issue. Numerical examples have illustrated that IRSOAR and IMSOAR generally works well but the former is more robust and can be considerably more efficient than the latter.

## References

- [1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. A. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, PA, 2000.
- [2] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur, NLEVP: A Collection of Nonlinear Eigenvalue Problems. Users' Guide, *MIMS EPrint 2010.98*, November 2010.
- [3] Z. Bai and Y. Su, SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem, *SIAM J. Matrix Anal. Appl.*, 26 (2005): 640–659.
- [4] Z. Bai and Y. Su, Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method, *SIAM J. Sci. Comput.*, 26 (2005): 1692–1709.
- [5] A. Feriani, F. Perotti, and V. Simoncini, Iterative system solvers for the frequency analysis of linear mechanical systems, *Computer Methods Appl. Mech. Engrg.*, 190 (2000): 1719–1739.
- [6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd Edition, The John Hopkins University, Baltimore, 1996.
- [7] N. J. Higham, R. C. Li and F. Tisseur, Backward error of polynomial eigenproblems solved by linearization, *SIAM J. Matrix Anal. Appl.*, 29 (2007): 1218–1241.
- [8] M. E. Hochstenbach, Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems, *BIT*, 44 (2004): 721–754.
- [9] H.-M. Huang, Z. Jia and W.-W. Lin, Convergence of q-Ritz pairs, refined q-Ritz vectors and q-Rayleigh-Ritz method for quadratic eigenvalue problems, arXiv: math/1109.6426v1, 2011.
- [10] Z. Jia, The convergence of generalized Lanczos methods for large unsymmetric eigenproblems, *SIAM J. Matrix Anal. Appl.*, 16 (1995): 643–862.
- [11] Z. Jia, Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems, *Linear Algebra Appl.*, 259 (1997): 1–23.

- [12] Z. Jia, Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and implicitly restarted refined Arnoldi algorithm, *Linear Algebra Appl.*, 287 (1999): 191–214.
- [13] Z. Jia, A refined subspace iteration algorithm for large sparse eigenproblems, *Appl. Numer. Math.*, 32 (2000): 35–52.
- [14] Z. Jia, The refined harmonic Arnoldi method and an implicitly restarted refined algorithm for computing interior eigenpairs of large matrices, *Appl. Numer. Math.*, 42 (2002): 489–512.
- [15] Z. Jia, Using cross-product matrices to compute the SVD, *Numer. Algor.*, 42 (2006): 31–61.
- [16] Z. Jia, Some theoretical comparisons of refined Ritz vectors and Ritz vectors, *Science in China Series A*, 47 (Suppl.) (2004): 222–233.
- [17] Z. Jia and D. Niu, An implicitly restarted refined bidiagonalization Lanczos method for computing a partial singular value decomposition, *SIAM J. Matrix Anal. Appl.*, 25 (2003): 246–265.
- [18] Z. Jia and D. Niu, A refined harmonic Lanczos bidiagonalization method and an implicitly restarted algorithm for computing the smallest singular triplets of large matrices, *SIAM J. Sci. Comput.*, 32 (2010): 714–744.
- [19] Z. Jia and G. W. Stewart, The Rayleigh–Ritz method for approximating eigenspaces, *Math. Comput.*, 270 (2001): 637–647.
- [20] E. Kokiopoulou, C. Bekas and E. Gallopoulos, Computing smallest singular triplets with implicitly restarted Lanczos bidiagonalization, *Appl. Numer. Math.*, 49 (2004): 39–61.
- [21] K. Meerbergen, The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem, *SIAM J. Matrix Anal. Appl.*, 34 (2008): 1463–1482.
- [22] C. Otto, Arnoldi and Jacobi–Davidson Methods for Quadratic Eigenvalue Problems, Diploma thesis, Institut für Mathematik, Technische Universität Berlin, Germany, 2004.
- [23] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Algorithms and Architectures for Advanced Scientific Computing, Manchester University Press, Manchester, UK, 1992.
- [24] D. C. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, *SIAM J. Matrix Anal. Appl.*, 13 (1992): 357–385.
- [25] G. W. Stewart, *Matrix Algorithms, Vol II: Eigensystems*, SIAM, Philadelphia, PA, 2001.
- [26] F. Tisseur, Backward error analysis of polynomial eigenvalue problems, *Linear Algebra Appl.*, 309 (2000): 339–361.
- [27] F. Tisseur and K. Meerbergen, The quadratic eigenvalue problem, *SIAM Rev.*, 43 (2001): 235–286.
- [28] H. A. Van der Vorst, *Computational Methods for Large Eigenvalue Problems*, Elsevier, North-Hollands, 2002.
- [29] S. Wei and I. Kao, Vibration analysis of wire and frequency response in the modern wiresaw manufacturing process, *J. Sound Vibration*, 231 (2000): 1383–1395.